

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

24. 1. 2005

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 4 年 1 月 3 0 日

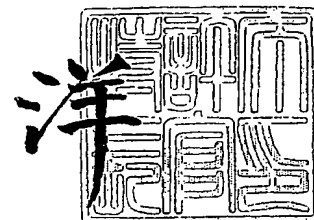
出 願 番 号  
Application Number: 特 願 2 0 0 4 - 0 2 4 4 8 5  
[ST. 10/C]: [J P 2 0 0 4 - 0 2 4 4 8 5]

出 願 人  
Applicant(s): キヤノン株式会社

2 0 0 5 年 3 月 1 0 日

特許庁長官  
Commissioner,  
Japan Patent Office

小 川



BEST AVAILABLE COPY

【書類名】 特許願  
【整理番号】 5514141-01  
【提出日】 平成16年 1月30日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 7/00  
【発明者】  
    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
    【氏名】 御手洗 裕輔  
【発明者】  
    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
    【氏名】 真継 優和  
【発明者】  
    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
    【氏名】 森 克彦  
【発明者】  
    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
    【氏名】 森江 隆  
【特許出願人】  
    【識別番号】 000001007  
    【氏名又は名称】 キヤノン株式会社  
【代理人】  
    【識別番号】 100076428  
    【弁理士】  
    【氏名又は名称】 大塚 康德  
【選任した代理人】  
    【識別番号】 100112508  
    【弁理士】  
    【氏名又は名称】 高柳 司郎  
【選任した代理人】  
    【識別番号】 100115071  
    【弁理士】  
    【氏名又は名称】 大塚 康弘  
【選任した代理人】  
    【識別番号】 100116894  
    【弁理士】  
    【氏名又は名称】 木村 秀二  
【手数料の表示】  
    【予納台帳番号】 003458  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 0102485

## 【書類名】特許請求の範囲

## 【請求項 1】

$n$  個のデータで構成される被符号化データを、 $m$  個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記適応化基底から、 $n$  個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

## 【請求項 2】

$n$  個のデータで構成される被符号化データを、少なくとも 2 つの、 $p$  個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記少なくとも 2 つの適応化基底から、 $n$  個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により、前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記少なくとも 2 つの適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

## 【請求項 3】

$n$  個のデータで構成される被符号化データを、少なくとも 2 つの、 $p$  個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により 1 つの前記適応化基底から、 $n$  個のデータからなる要素復元データを生成する復元ステップと、

所定の誤差評価関数により、差分被符号化データと前記要素復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 つの適応化基底を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

## 【請求項 4】

被符号化データである  $k$  次元空間分布データ  $f(x_1, x_2, \dots, x_k)$  を、請求項 3 に記載の前記被符号化データを用いて、少なくとも 1 組の、 $k$  個の 1 次元適応化基底組  $X_1(x_1)$ 、 $X_2(x_2)$ 、 $\dots$   $X_k(x_k)$  を生成し、

$$f(x_1, x_2, \dots, x_k) = \sum \prod_k X_k(x_k)$$

と展開表現することを特徴とするデータ符号化方法。

## 【請求項 5】

被符号化データである  $k$  次元空間分布データ  $f(x_1, x_2, \dots, x_k)$  を、 $k$  個の 1 次元適応化基底組  $X_1(x_1)$ 、 $X_2(x_2)$ 、 $\dots$   $X_k(x_k)$  に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

前記 1 次元適応化基底組から、復元データを次式で生成する復元ステップと、

$$\tilde{f}(x_1, x_2, \dots, x_k) = \prod_k X_k(x_k)$$

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 6】

被符号化データである  $k$  次元空間分布データ  $f(x_1, x_2, \dots, x_k)$  を、少なくとも 2 組の、 $k$  個の 1 次元適応化基底組 に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

前記 1 次元適応化基底組から、復元データを次式で復元する復元ステップと、

$$\tilde{f}(x_1, x_2, \dots, x_k) = \sum \prod X_u(x_k)$$

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 7】

被符号化データである  $k$  次元空間分布データ を、少なくとも 2 組の、 $k$  個の 1 次元適応化基底組 に符号化する方法であって、

前記 1 次元適応化基底組を初期化するステップと、

1 組の前記 1 次元適応化基底組から、要素復元データを次式で生成する復元ステップと

$$\tilde{f}_i(x_1, x_2, \dots, x_k) = \prod X_u(x_k)$$

所定の誤差評価関数により、差分被符号化データと前記要素復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記 1 組の 1 次元適応化基底組を修正する修正ステップと

を有することを特徴とするデータ符号化方法。

【請求項 8】

前記復元ステップは、前記少なくとも 2 つの適応化基底の各々について、所定の変換で  $n$  個のデータで構成される要素復元データに変換し、前記変換された要素復元データの線形和をとり、前記復元データとすることを特徴とする請求項 2 に記載のデータ符号化方法。

【請求項 9】

前記算出した誤差の収束を判定するステップを更に有し、前記誤差が収束したと判定されるまで、前記修正ステップを繰り返し行うことを特徴とする請求項 1、2、5、または 6 のいずれか 1 つに記載のデータ符号化方法。

【請求項 10】

前記算出した誤差の収束を判定するステップを更に有し、

前記誤差が収束したと判定されるまで、前記修正ステップを繰り返し行い、前記誤差が収束したと判定された場合に、次の新たな適応化基底、若しくは新たな 1 次元適応化基底組に対し、前記修正ステップを行う

ことを特徴とする請求項 3、または請求項 7 のいずれか 1 項に記載のデータ符号化方法。

【請求項 11】

前記差分被符号化データは、前記被符号化データから、前記誤差の収束を判定するステップにより、前記誤差が収束したと判定された全ての前記適応化基底、若しくは前記 1 次元適応化基底組から復元される前記要素復元データを差し引いたものであることを特徴とする請求項 10 に記載のデータ符号化方法。

【請求項 12】

前記修正ステップを、前記算出した誤差が予め定めた目標誤差以下になるまで繰り返し行うことを特徴とする請求項 1 乃至請求項 3、請求項 5 乃至請求項 7 のいずれか 1 項に記

載の記載のデータ符号化方法。

【請求項 13】

前記誤差評価関数は、前記符号化データ、若しくは前記差分被符号化データと、前記復元データ、若しくは前記要素復元データの各要素との2乗誤差の総和であることを特徴とする請求項1乃至請求項3、または請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 14】

前記修正ステップにおいて、前記算出した誤差が減少するように、前記適応化基底、若しくは前記1次元適応化基底組の修正を行うことを特徴とする請求項1乃至請求項3、または請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 15】

前記修正ステップにおいて、前記適応化基底、若しくは前記1次元適応化基底組の修正量は、前記誤差評価関数の、前記適応化基底の内の修正する要素、若しくは前記1次元適応化基底組の内の修正する要素での偏微分係数に、所定の係数を乗じたものであることを特徴とする請求項1乃至請求項3、または請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 16】

前記所定の係数は、負の定数であることを特徴とする請求項15に記載のデータ符号化方法。

【請求項 17】

前記所定の係数は、負の値であり、徐々に0に近づけることを特徴とする請求項15に記載のデータ符号化方法。

【請求項 18】

前記被符号化データを、所定範囲で分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項1乃至請求項3、または請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 19】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 20】

前記被符号化データを、周波数空間に変換し符号化を行うことを特徴とする請求項5乃至請求項7のいずれか1項記載のデータ符号化方法。

【請求項 21】

前記被符号化データを、周波数空間に変換し、周波数空間において所定範囲に分割し、前記所定範囲ごとに符号化を行うことを特徴とする請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 22】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに周波数空間に変換し符号化を行うことを特徴とする請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 23】

前記被符号化データを、空間的に所定範囲で分割し、前記所定範囲ごとに周波数空間に変換し、前記周波数空間に変換された所定範囲ごとの所定周波数成分を周波数データとして合成し、前記周波数データの符号化を行うことを特徴とする請求項5乃至請求項7のいずれか1項に記載のデータ符号化方法。

【請求項 24】

請求項1、5、23のいずれか1項に記載のデータ符号化方法により生成される前記適応化基底、若しくは前記1次元適応化基底組から、前記所定の変換により復元データを復元することで、復号化することを特徴とするデータ復号化方法。

【請求項 25】

請求項 2、3、6、7、23 のいずれか 1 項に記載のデータ符号化方法により生成される前記適応化基底、若しくは前記 1 次元適応化基底組から、前記所定の変換により要素復元データを復元し、前記復元した要素復元データの線形和をとることで、復号化することとを特徴とするデータ復号化方法。

【請求項 26】

k 次元荷重分布データ  $w(x_1, x_2, \dots, x_k)$  と、k 次元分布データである被積和演算データ  $A(x_1, x_2, \dots, x_k)$  とを次式で積和演算を行う積和演算方法であって、

$$y = \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot w(x_1, x_2, \dots, x_k) dx_1 dx_2 \dots dx_k$$

前記 k 次元荷重分布データを、請求項 5 乃至 7 のいずれか 1 項に記載のデータ符号化方法により、少なくとも 1 組の、k 個の 1 次元適応化基底組  $X_{11}(x_1)$ 、 $X_{21}(x_2)$ 、 $\dots$ 、 $X_{k1}(x_k)$  に符号化し、

前記被積和演算データと、前記 1 次元適応化基底組を用いて、前記積和演算を次式で求める

$$y = \sum \left[ \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot \prod_i X_{ki}(x_k) dx_1 dx_2 \dots dx_k \right]$$

ことを特徴とする積和演算方法。

【請求項 27】

n 個のデータで構成される被符号化データを、m 個のデータで構成される適応化基底に符号化する装置であって、

前記適応化基底を初期化する手段と、

所定の変換により前記適応化基底から、n 個のデータからなる復元データを生成する復元手段と、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出手段と、

前記誤差に基づき前記適応化基底を修正する修正手段を有することを特徴とするデータ符号化装置。

【請求項 28】

k 次元の被符号化データを、k 個の一次元適応化基底の組に符号化するデータ符号化方法であって、

被符号化データを所定のメモリに記憶する記憶ステップと、

前記 k 個の一次元適応化基底をメモリに確保し、初期化する初期化ステップと、

所定の変換により、前記メモリ内の一次元適応化基底から、k 次元の復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

該誤差演算ステップで得られた誤差に基づき、前記適応化基底を修正するか否かを判定し、前記適応化基底を修正すると判定した場合には前記メモリの適応化基底を修正して、前記復元ステップ、誤差演算ステップを繰り返し、前記適応化基底を修正しないと判定した場合には、最後の適応化基底を符号化結果として出力する判定ステップと

を備えることを特徴とするデータ符号化方法。

【請求項 29】

前記判定ステップは、誤差が収束したか否かに基づいて適応化基底を修正するか否かを判定し、

収束したと判定した場合であって、被符号化データと前記復元データとの誤差が所定値以上の差を有する場合には、k 個の一次元適応化基底の組を更に追加することを特徴とする請求項 28 に記載のデータ符号化方法。

【請求項 30】

k 次元の被符号化データを、k 個の一次元適応化基底の組に符号化するデータ符号化方

法であって、

被符号化データを所定のメモリに記憶する被符号化データ記憶手段と、  
前記 k 個の一次元適応化基底を記憶する基底記憶手段と、  
該基底記憶手段に記憶された k 個の一次元適応化基底を初期化する初期化手段と、  
所定の変換により、前記基底記憶手段に記憶された k 個の一次元適応化基底から、k 次元の復元データを生成する復元手段と、  
所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出手段と、  
該誤差演算手段で得られた誤差に基づき、前記適応化基底を修正するか否かを判定し、前記適応化基底を修正すると判定した場合には前記基底記憶手段に記憶された適応化基底を修正して、前記復元手段、誤差演算手段を繰り返し、前記適応化基底を修正しないと判定した場合には、最後の適応化基底を符号化結果として出力する判定手段と、  
を備えることを特徴とするデータ符号化装置。

【請求項 31】

k 次元荷重分布データ  $w(x_1, x_2, \dots, x_k)$  と、k 次元分布データである被積和演算データ  $A(x_1, x_2, \dots, x_k)$  とを次式で積和演算を行う積和演算装置であって、

$$y = \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot w(x_1, x_2, \dots, x_k) dx_1 dx_2 \dots dx_k$$

前記 k 次元荷重分布データを、請求項 5 乃至 7 のいずれか 1 項に記載のデータ符号化方法により、少なくとも 1 組の、k 個の 1 次元適応化基底組  $X_{11}(x_1)$ 、 $X_{21}(x_2)$ 、 $\dots$ 、 $X_{k1}(x_k)$  に符号化し、

前記被積和演算データと、前記 1 次元適応化基底組を用いて、前記積和演算を次式で求める

$$y = \sum \left[ \iint \dots \int A(x_1, x_2, \dots, x_k) \cdot \prod_i X_{ki}(x_k) dx_1 dx_2 \dots dx_k \right]$$

ことを特徴とする積和演算装置。

【請求項 32】

前記積和演算装置において、積和演算を行なう演算素子がアレイ状に構成されていることを特徴とする請求項 31 記載の積和演算装置。

【請求項 33】

前記積和演算装置において、1 次元適応化基底組がアレイ状に構成された演算素子全てに☐されることを特徴とする請求項 32 に記載の積和演算装置。

【請求項 34】

前記積和演算装置において、アレイ状に構成された演算素子全てにおいて前記積和演算が実行されることを特徴とする請求項 33 に記載の積和演算装置。

【請求項 35】

前記積和演算装置において、1 次元適応化基底組が、アレイ状に構成された演算素子の列・行のそれぞれ一部に☐されることを特徴とする請求項 32 に記載の積和演算装置。

【請求項 36】

前記積和演算装置において、アレイ状に構成された演算素子の一部において前記積和演算が実行されることを特徴とする請求項 35 に記載の積和演算装置。

## 【書類名】明細書

【発明の名称】符号化方法及び装置、並びにコンピュータプログラム及びコンピュータ可読記憶媒体

## 【技術分野】

## 【0001】

本発明は、データの符号化、特に2次元画像や3次元密度分布等の多次元空間における分布データの符号化技術に関するものである。

## 【背景技術】

## 【0002】

2次元画像等の多次元分布データを符号化する技術としては、JEPG形式の圧縮技術のような、2次元画像データに対し離散コサイン変換を行い、変換されたデータに所定の量子化を行い、それをハフマン符号化等により符号化することで、データ量を圧縮する技術等が一般的である。

## 【0003】

また、画像データを予測符号化し、予測符号化による予測誤差データをブロック分割し、ブロック毎に予測誤差の大きいブロックか小さいブロックかを判別する手法もある（特許文献1）。この文献では、予測誤差の大きいブロックにはベクトル量子化を適用し、ベクトル量子化により検索されたコードブックのベクトル値と予測誤差との差分値を計算し、この差分値と予測誤差の小さいブロックにおける予測誤差をエントロピー符号化し、エントロピー符号とブロック大小判別フラッグとベクトル量子化によるインデックスとを用いて符号化データを作成するものである。このように、省スペースでのデータ保持、高速通信等の観点から、高効率なデータ符号化技術が望まれている。

## 【0004】

また、まったく別の観点ではあるが、画像処理等で多用される所定の2次元荷重分布データによる、2次元画像に対する離散的コンボリューション演算を行う演算器において、積和演算素子で大規模並列化し、高速な演算を可能にすることが期待されている。しかし配線の問題等から、積和演算を2次元的に並列化して実行することは困難であった。従って、コンボリューションの2次元荷重分布データを1次元の基底組に分解する方法が望まれている。

【特許文献1】特開平11-331852号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0005】

本発明は、このような点に鑑みなされたものであり、高効率なデータの符号化、特に2次元画像等の多次元空間における分布データ等を、幾つかの一次元の基底に分解し、符号化する技術を提供とするものである。

## 【課題を解決するための手段】

## 【0006】

この課題を解決するため、例えば本発明の符号化方法は以下の工程を備える。すなわち、

n個のデータで構成される被符号化データを、m個のデータで構成される適応化基底に符号化する方法であって、

前記適応化基底を初期化するステップと、

所定の変換により前記適応化基底から、n個のデータで構成される復元データを生成する復元ステップと、

所定の誤差評価関数により前記被符号化データと前記復元データとの誤差を算出する誤差算出ステップと、

前記誤差に基づき前記適応化基底を修正する修正ステップとを有する。

## 【発明の効果】

## 【0007】



本発明によれば、高効率なデータの符号化、特に2次元画像等の多次元空間における分布データ等を、高効率に符号化することができる。また2次元のデータを1次元の基底データに分解することが可能になるため、コンボリューション演算器において問題となっていた、2次元並列化による配線の問題を解決することができる。

【発明を実施するための最良の形態】

【0008】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【0009】

<第1の実施形態>

第1の実施形態として、有限範囲で定義される2次元濃淡画像を、2つの1次元適応化基底に符号化する方法を説明する。

【0010】

図8は本第1の実施形態における装置構成を示すブロック図である。

【0011】

図中、1は装置全体の制御を司るCPUであり、2はCPU1の処理手順（プログラム）を記憶しているROMである。3はCPU1のワークエリアとして使用されるRAMであり、ここには符号化対象の画像データを格納する画像データエリア3a、並びに、基底データを格納するエリア3bが確保される。なお、エリア3a、3bのサイズは入力する画像データサイズに応じて決定されることになる。4は符号化対象の濃淡画像データを入力する画像入力部である。画像入力部としては、イメージスキャナ、撮像素子等であるが、画像データが記憶媒体に記憶されている場合にはその記憶媒体をアクセスする装置であってもよいし、ネットワークを介して画像を受信する場合にはネットワークインタフェースであっても構わない。5は符号化後のデータを格納する外部記憶装置（ハードディスク等）であるが、符号化データをネットワーク上に出力する場合にはネットワークインタフェースでも構わない。

【0012】

実施形態では説明を簡単なものとするため、符号化対象の画像データは1画素8ビットの多値（濃淡）モノクロ画像として説明する。また、画像は矩形であり、水平方向の画素数が $x_{max}$ 、垂直方向の画素数が $y_{max}$ であるものとする。画像入力部4がイメージスキャナであれば、読取り解像度、原稿のサイズにより、 $x_{max}$ 、 $y_{max}$ は一義的に決まるものである。

【0013】

入力した画像は画像データエリア3aに格納されるが、座標 $(x, y)$ （但し、 $0 \leq x < x_{max}$ 、 $0 \leq y < y_{max}$ ）の画素値を便宜的に $f(x, y)$ と表現する。また、入力する画像データのサイズが決定されるので、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ を基底エリア3bに確保する。 $x$ 、 $y$ の取り得る範囲は入力画像のそれと同じである。なお、画素値 $f(x, y)$ は8ビットの整数値であるが、1次元適応化基底 $X(x)$ 、 $Y(y)$ は小数点の値を記憶保持できるように多バイト形式でRAM3b内の基底エリア3bに確保することになる。

【0014】

以下、本第1の実施形態におけるCPU1の符号化処理手順を図1のフローチャート（対応するプログラムはROM2に格納されている）に従って説明する。

【0015】

まず、画像入力（ステップS10）において、符号化される2次元濃淡画像である画像 $f(x, y)$ を入力し、画像エリア3aに格納する。次に、基底設定を行う（ステップS11）。ここでは、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ について初期値を設定する。但し、この1次元適応化基底 $X(x)$ 、 $Y(y)$ は、 $0 \leq x < x_{max}$ 、 $0 \leq y < y_{max}$ という有限範囲において「0」以外の値を設定する。初期段階では乱数関数を用いて設定することになる。なお、ここでは上記のように各定義域を有限としたが、特にこれに限るものではない。すなわち、無限の定義域において定義された $f(x, y)$ 、 $X(x)$ 、 $Y(y)$

(y) においても、後述の誤差 E、及び修正量  $\Delta X(x)$ 、 $\Delta Y(y)$  が発散しない限り、本手法は適用可能である（但し、メモリ容量は有限であるので、自ずとその限界はある）。

【0016】

基底復元（ステップ S12）では、復元画像  $f^-(x, y)$  を次のように生成する。

【数1】

$$\tilde{f}(x, y) = X(x) \cdot Y(y)$$

【0017】

次いで、ステップ S13 の誤差演算処理にて、入力画像  $f(x, y)$  と復元画像  $f^-(x, y)$  との誤差 E を、次の誤差評価関数を用いて算出する。

【数2】

$$E = \iint \frac{1}{2} \{f(x, y) - \tilde{f}(x, y)\}^2 dx dy$$

ここで、 $\int f(x) dx$  は、 $x$  の定義領域全範囲における  $f(x)$  の積分を示す。 $y$  成分についても同様に積分することになるので、誤差 E は入力画像全域における累積誤差となる。また、この誤差 E は、後述する誤差収束判定（分岐判定であるステップ S16）において用いるため、履歴として RAM3 に順次記録しておく。

【0018】

続いて、この誤差 E に基づいて、修正量の算出を行う（ステップ S14）。ここでは、各 1 次元適応化基底の修正量  $\Delta X(x)$ 、 $\Delta Y(y)$  を以下のように算出する。

【数3】

$$\Delta X(x) = -\eta_x \cdot \frac{\partial E}{\partial X(x)} = \eta_x \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot Y(y) dy$$

$$\Delta Y(y) = -\eta_y \cdot \frac{\partial E}{\partial Y(y)} = \eta_y \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot X(x) dx$$

ここで、 $\eta_x$ 、 $\eta_y$  は修正パラメータであり正の定数である（この  $\eta_x$ 、 $\eta_y$  の設定については後述する）。

【0019】

この修正量を得る関数は、誤差 E を求める式の偏微分式であることに注意されたい。すなわち、誤差 E は常に 0 以上の値を有するわけであり、究極的には、本実施形態の目標は誤差 E が極小値となる 1 次元適応化基底を求める点にある。現時点での 1 次元適応化基底による傾きがマイナスであるということは、極小値となる一次元基底は現時点より正の方向に位置することになり、逆に、1 次元適応化基底による傾きがプラスであるということは、極小値となる一次元基底は現時点より負の方向に存在することを意味する。それ故、傾きに対してマイナスを乗算することで、傾きに応じた修正量を算出することになる。

【0020】

次に、処理はステップ S15 に進み、基底修正処理を行う。これは、ステップ S14 にて得られた修正量  $\Delta X(x)$ 、 $\Delta Y(y)$  を修正前の 1 次元適応化基底を  $X(x)$ 、 $Y(y)$  に加算することで、修正後の 1 次元適応化基底を  $X'(x)$ 、 $Y'(y)$  を求めるものである。式で示せば次の通りである。

$$X'(x) = X(x) + \Delta X(x)$$

$$Y'(y) = Y(y) + \Delta Y(y)$$

この 1 次元適応化基底の修正の際には、入力画像  $f(x, y)$  が、定義された全ての  $x$ 、 $y$  の範囲において常に 0 でない限り、各 1 次元適応化基底が常に 0 ではないという上記条件を満たさなければならない。ただし、そのような条件になってしまった場合は、1 次元適応化基底の少なくとも一部にランダムに微小の揺らぎを与えるようにすればよい。

## 【0021】

上記のステップS12（基底復元処理）からステップS15（基底修正処理）までの1次元適応化基底の修正を、修正された1次元適応化基底に対して、誤差下限判定（ステップS16）、及び誤差収束判定（ステップS17）において終了判定が出されるまで繰り返し実行する。

## 【0022】

ここで、ステップS16の誤差下限判定は、誤差算出（ステップS13）で算出した、直前の修正の誤差Eが、予め定めておいた所定の値以下であった場合に、その時の1次元適応化基底が許容範囲の誤差Eを与える1次元適応化基底であるか否かを判定するものである。そして、終了判定の場合には、その時点での1次元適応化基底を符号化データとして記録して処理を終了する。

## 【0023】

また、ステップS17の誤差収束判定は、誤差算出（ステップS13）で算出し、記録した誤差Eの履歴から、誤差の収束判定を行い、誤差が収束したと判定された場合は、その時の1次元適応化基底が誤差の極小値を与える1次元適応化基底であるとして終了判定する。収束したと判断した場合の1次元適応化基底は符号化データとして記録して処理を終了する。誤差の収束の判定としては、1回の修正ごとの、誤差Eの変化がほとんどない状況（誤差の差分が所定閾値以下の場合）が続いた場合に、誤差が収束したと判定する。

## 【0024】

上記説明した処理により、2次元濃淡画像 $f(x, y)$ を、2つの1次元適応化元基底 $X(x)$ 、 $Y(y)$ に符号化することができる。換言すれば、画像データ $f(x, y)$ は画素数は $x_{\max} \times y_{\max}$ 画素存在していたのに対し、符号化データは $x_{\max} + y_{\max}$ 個のデータにまで圧縮されたことになる（ただし、入力画像の1画素は8ビットであるのに対し、1次元適応化元基底の1要素は数バイト必要になる）。また、この符号化された2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ の復号化画像 $f_{\text{decode}}(x, y)$ は、次のようにして復号できる。

$$f_{\text{decode}}(x, y) = X(x) \times Y(y)$$

上記符号化処理において、基底設定（ステップS11）において設定する初期の1次元適応化基底 $X(x)$ 、 $Y(y)$ としては、定義された $x$ 、 $y$ の範囲において $X(x)$ 、 $Y(y)$ は入力画像のレンジ程度の値であることが好ましい。好適には、0ではないものの、ほぼ0に近い正の値となるように設定すると、初期値依存性が少なくなり、適切な1次元適応化基底を得られる可能性が高くなる。

## 【0025】

また、修正パラメータ $\eta_x$ 、 $\eta_y$ は、入力画像 $f(x, y)$ のレンジや定義された $x$ 、 $y$ の範囲の幅に応じて定めるのが好ましい。具体的には $\eta_x$ は $y$ の範囲の2乗と $f(x, y)$ のレンジに反比例し、 $\eta_y$ は $x$ の範囲の2乗と $f(x, y)$ のレンジに反比例させるようにするのが好適である。ここでは修正パラメータを $\eta_x$ 、 $\eta_y$ と各1次元適応化基底により異ならせているが、定義された $x$ 、 $y$ の範囲の幅が同程度であれば、同一の値を用いても構わない。この修正パラメータを大きい値に設定すると、少ない回数の1次元適応化基底の修正で誤差下限判定（ステップS16）、若しくは誤差収束判定（ステップS17）での終了判定が出る可能性が高くなるが、誤差Eが極小値周辺で大きく振動し、適切な1次元適応化基底を得られない可能性が高くなる。また、小さい値に設定すると、誤差Eが極小値付近に落ち着いて、適切な1次元適応化基底を得られる可能性が高くなるが、誤差下限判定（ステップS16）、若しくは誤差収束判定（ステップS17）での終了判定までに、多くの回数の、1次元適応化基底の修正を必要とする可能性が高くなる。上記の点を鑑み、修正パラメータは、誤差Eが極小値周辺で大きく振動しない程度に小さく、極端に多くの回数の、1次元適応化基底の修正を必要としない程度の大きな値にしておくことが好適である。また修正パラメータを修正回数に応じて徐々に小さくしていく手法を適用しても良い。

## 【0026】

また、誤差下限判定（ステップS16）で用いる、予め定めておく誤差Eの許容範囲も、入力画像 $f(x, y)$ のレンジや、定義された $x, y$ の範囲の幅に応じて定めるのが好ましく、これを大きい値にすると、少ない回数の1次元適応化基底の修正で誤差下限判定（ステップS16）での終了判定が出る可能性が高くなるが、誤差Eが高い1次元適応化基底を得る可能性が高くなり、小さい値にすると、誤差が小さい適切な1次元適応化基底を得られる可能性が高くなる。しかし、誤差下限判定（ステップS16）での終了判定までに、多くの回数の、1次元適応化基底の修正を必要とする、若しくは誤差下限判定（ステップS16）での終了判定が出ない可能性が高くなる。

## 【0027】

さらに、上記説明した修正量算出（ステップS14）で算出する修正量 $\Delta X(x)$ 、 $\Delta Y(y)$ を、予め保持しておいた、この修正の前の修正量 $\Delta X^{\wedge}(x)$ 、 $\Delta Y^{\wedge}(y)$ を用いて、以下のようにすることで、より誤差の小さい好適な1次元適応化基底を得られる、若しくは少ない修正回数で所望の1次元適応化基底を得られる可能性が高くなる。

## 【数4】

$$\Delta X(x) = \eta_x \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot Y(y) dy + \alpha_x \cdot \Delta \hat{X}(x)$$

$$\Delta Y(y) = \eta_y \cdot \int \{f(x, y) - \tilde{f}(x, y)\} \cdot X(x) dx + \alpha_y \cdot \Delta \hat{Y}(y)$$

ここで $\alpha_x$ 、 $\alpha_y$ は、慣性項係数であり、1未満の正の定数である。ここでは、慣性項係数を各1次元適応化基底により異ならせているが、同一の値を使っても構わない。ここで用いる慣性項は、大きい値に設定する程、上記利点が発生する可能性が高くなるが、上記利点と逆の効果も現れ、1次元適応化基底の修正が不安定になる可能性が高くなるため、あまり大きな値に設定することは好ましくない。また小さい値に設定すると、上記利点の逆の効果が現れる可能性は低くなるが、上記利点を得られる可能性は少なくなる。そのため、この慣性項係数は、1次元適応化基底の修正が不安定にならないような範囲でできる限り大きな値に設定することが好ましい。

## 【0028】

以上、第1の実施形態では、有限範囲で定義される2次元濃淡画像 $f(x, y)$ を、2つの1次元適応化基底 $X(x)$ 、 $Y(y)$ に符号化する方法の例として説明したが、符号化対象は2次元に限らない。例えば3次元の空間における密度分布等にも適用可能であるし、それを越える多次元に対しても同様に適用可能である。

## 【0029】

## &lt;第2の実施形態&gt;

上記第1の実施形態では、符号化される2次元画像が比較的単純なものであれば、座標系を適切に選択することにより、誤差の少ない符号化が可能である。つまり、例えば2次元画像が、図5に示すような単純な矩形形状のものであれば、その矩形に直交する直交座標系を用いて符号化すれば、比較的誤差の少ない符号化が可能になる。このようにデータに応じて極座標系等や、3次元データであれば円柱座標系等を適切に選択してやることで、誤差Eが少ない符号化が可能である。

## 【0030】

しかし、2次元画像が例えば自然画像であるような場合は、上記第1の実施形態で示した方法では、符号化されたデータを用いて復号したデータ（画像データ）と、元のデータの誤差が大きくなってしまう。そこで、本第2の実施形態では、上記第1の実施形態の変形例として、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、複数の離散的な1次元適応化基底組 $X_i(x)$ 、 $Y_i(y)$ （ $x, y$ は整数）に符号化する方法を説明する。

## 【0031】

概要を簡単に説明すると、1つの1次元適応化基底を用意し、第1の実施形態と同様の処理を行って誤差Eの下限判定、並びに、収束判定を行った後、誤差Eが十分に小さいと

は判断されない場合には、新たな一次元適応化基底組を追加し残された被符号化データと復元データの誤差を小さくするように、再度、同じ処理を繰り返すものである。

【0032】

装置構成は、第1の実施形態と同じとし、以下、本第2の実施形態に係る処理を図2のフローチャートに従って説明する。

【0033】

まず、画像入力（ステップS20）において、符号化される離散的な2次元濃淡画像である入力画像  $f(x, y)$  を入力する。ここでも第1の実施形態と同様に、 $x, y$  は所定の定義域をもち、 $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$  とする。また、本第2の実施形態においては、入力画像は離散的であり、 $x, y$  は整数である。次に、基底設定処理を行う（ステップS21）。ここでは、離散的な1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  ( $x, y$  は整数) を1つ選択して初期の1次元適応化基底組として設定する。添字1は基底番号であり、特に限定はしないが、本第2の実施形態では0から順に、予め定めた基底組数まで順に選択していくことにする。

【0034】

次の基底復元処理を行う（ステップS22）。ここでは、選択した基底番号1の1次元適応化基底組から、要素復元画像  $f_1^-(x, y)$  を次のように算出する。

【数5】

$$\tilde{f}_1(x, y) = X_1(x) \cdot Y_1(y)$$

【0035】

次に、誤差算出（ステップS23）により、誤差  $E$  を以下のように算出する。

【数6】

$$E = \sum_x \sum_y \left[ \frac{1}{2} \{ f_1(x, y) - \tilde{f}_1(x, y) \}^2 \right]$$

上式において、 $f_1(x)$  は差分被符号化画像であり、以下のように表される。

【数7】

$$f_1(x, y) = f(x, y) - \sum_{n=0}^{l-1} \tilde{f}_n(x, y)$$

ここで、

【数8】

$$\sum_x f(x)$$

は、 $x$  の定義域内全範囲における、 $x$  が整数の時の  $f(x)$  の総和を意味する。

【0036】

また、

【数9】

$$\sum_{x=n}^m f(x)$$

は、 $n \leq x \leq m$  の  $f(x)$  の総和を意味し、 $m < n$  の場合は0となる。

【0037】

上記の差分被符号化画像  $f_1(x, y)$  は、入力画像  $f(x, y)$  から、基底番号0から基底番号1-1までの、1組の1次元適応化基底組により復元される各要素復元画像を差し引いたものを意味する。つまり、本第2の実施形態では基底番号0から順に1次元適応化基底組を選択していき、各1次元適応化基底組を修正していくので、差分被符号化画像  $f_1(x, y)$  は、入力画像  $f(x, y)$  から、既に修正完了した1次元適応化基底組により復元される各要素復元画像を差し引いたものとなる。

## 【0038】

次いで、修正量算出（ステップS24）において、基底設定（ステップS21）において選択された1次元適応化基底組 $X_1(x)$ 、及び $Y_1(y)$ に対する修正量 $\Delta X_1(x)$ 、及び $\Delta Y_1(y)$ を以下のように算出する。

## 【数10】

$$\Delta X_1(x) = -\eta_x \cdot \frac{\partial E}{\partial X_1(x)} = \eta_x \cdot \sum_y \{f_1(x, y) - \tilde{f}_1(x, y)\} \cdot Y_1(y)$$

$$\Delta Y_1(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_1(y)} = \eta_y \cdot \sum_x \{f_1(x, y) - \tilde{f}_1(x, y)\} \cdot X_1(x)$$

上式において $\eta_x$ 、 $\eta_y$ は修正パラメータであり、第1の実施形態で説明したものと同様のものである。

## 【0039】

上記算出した修正量 $\Delta X_1(x)$ 、及び $\Delta Y_1(y)$ を用い、基底修正（ステップS25）において、基底設定（ステップS21）で選択された1次元適応化基底組 $X_1(x)$ 、 $Y_1(y)$ に対して、修正後の各1次元適応化基底組を $X'_1(x)$ 、 $Y'_1(y)$ 、修正前の1次元適応化基底組を $X_1(x)$ 、 $Y_1(y)$ とした時に、各1次元適応化基底組を次のように修正する。

## 【数11】

$$X'_1(x) = X_1(x) + \Delta X_1(x)$$

$$Y'_1(y) = Y_1(y) + \Delta Y_1(y)$$

## 【0040】

上記修正において、第1の実施形態で示した前回の修正量を慣性項係数により重み付けして付加する方法を用いても構わない。このような修正を、第1の実施形態と同様に、誤差下限判定（ステップS26）において終了判定、若しくは誤差収束判定（ステップS27）において収束判定がなされるまで繰り返し行うことで、基底設定（ステップS21）で選択した基底番号の1次元適応化基底組が修正される。

## 【0041】

誤差下限判定（ステップS26）、及び誤差収束判定（ステップS27）での判定処理は、第1の実施形態で示したものと同様である。誤差下限判定（ステップS26）において、終了判定が出た場合は、ここまでに選択され修正された基底組数と、それらの1次元適応化基底組を符号化データとして記録して処理を終了する。終了判定が出なかった場合は、次の誤差収束判定（ステップS27）に進み、ここで誤差Eの収束判定を行う。

## 【0042】

第1の実施形態では、誤差収束判定（ステップS27）において、誤差Eが収束したと判定された場合は、終了判定を出し処理を終了したが、本第2の実施形態では、次の処理である差分算出（ステップS28）の処理に進む。収束したと判定されない場合は、基底復元（ステップS22）に戻り、選択された基底番号の1次元適応化基底組に対する修正を繰り返し実行する。

## 【0043】

さて、誤差Eが収束したと判定された場合、差分算出（ステップS28）において、基底番号1の1次元適応化基底組から復元される要素復元画像 $f^{-1}_1(x, y)$ を

## 【数12】

$$\tilde{f}_1(x, y) = X_1(x) \cdot Y_1(y)$$

で算出し、差分被符号化画像 $f_{1+1}(x, y)$ を、

## 【数 1 3】

$$f_{l+1}(x, y) = f(x, y) - \sum_{n=0}^l \tilde{f}_n(x, y)$$

により生成する。

## 【0044】

次に、基底数判定（ステップ S 2 8）において、ここまでに修正が完了した基底組数が、予め定めた所定数に達していた場合は、終了判定を出し、ここまでに選択され修正された基底組数と、それらの 1 次元適応化基底組を符号化データとして記録して処理を終了する。基底組数が所定数（許容数）に達していない場合は、基底設定（ステップ S 2 1）に戻り、新たな 1 次元適応化基底組を選択して、上記説明した基底復元（ステップ S 2 2）から誤差収束判定（ステップ S 2 7）までの処理を繰り返し実行し、新たに選択された 1 次元適応化基底組の修正を行う。

## 【0045】

基底数判定処理（ステップ S 2 8）において用いる、予め定めた基底組の所定数は、無限個と設定しておいても構わないが、そのようにすると、誤差下限判定（ステップ S 2 6）で用いる誤差の許容範囲を仮に小さく設定していると、非常に多数の基底組数を必要とする可能性が高くなる。そこで、基底組数に予め制限を与えておくことにより、データ量の上限を定めて上記符号化を行うことができる。

## 【0046】

上記方法により符号化した符号化データを復号化画像  $f_{\text{decode}}(x, y)$  に復号するのは、下記に示すように行えばよい。

## 【数 1 4】

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} [X_n(x) \cdot Y_n(y)]$$

上式の  $k$  は選択され修正された基底組数を意味する。このように、本発明の符号化方法では、簡単な処理により、符号化された符号化データから、復号化データを得ることができる特徴がある。

## 【0047】

以上説明したように第 2 の実施の形態では、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数の離散的な 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に符号化する。この結果、第 1 の実施形態と比較して、符号化対象が自然画の場合にも、精度良く対処することが可能となる。

## 【0048】

## &lt; 第 3 の実施の形態 &gt;

本第 3 の実施形態では、上記第 2 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数の離散的な 1 次元適応化基底組  $X^{\varepsilon, \delta}_1(x)$ 、 $Y^{\varepsilon, \delta}_1(y)$  に変換し、それを符号化する方法を第 3 の実施形態として説明する。また、装置構成は第 1 の実施形態と同じであるものとし、その説明は省略する。

## 【0049】

図 3 は、本第 3 の実施形態における処理のフローを示したものである。以下、同図に従って本第 3 の実施形態における処理手順を説明する。

## 【0050】

まず、画像入力（ステップ S 3 0）において、符号化される離散的な 2 次元濃淡画像である入力画像  $f(x, y)$  を入力する。第 1、第 2 の実施形態と同様、 $x$ 、 $y$  の定義域は  $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$  であり、共に整数である。

## 【0051】

次に、画像分割処理（ステップ S 3 1）において、入力画像をサイズ  $L_x$ 、 $L_y$ （ $L_x$ 、 $L_y$  は非負整数）の、以下の式で表される複数のウィンドウ  $W^{\varepsilon, \delta}(x, y)$  に分割

する。

【数 15】

$$W^{\xi, \zeta}(x, y) = f(x + \xi L_x, y + \zeta L_y)$$

ここで、 $\xi, \zeta$  は 0 以上の整数であり、各ウィンドウの定義域は  $0 \leq x < L_x$ 、 $0 \leq y < L_y$  である。また、 $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$  となるウィンドウでは、 $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$  の範囲を切り捨ててしまい  $L_x$ 、 $L_y$  よりサイズの小さいウィンドウとすればよい。

【0052】

次に、ウィンドウ選択処理を行う（ステップ S 3 2）。ここでは、画像分割処理（ステップ S 3 1）で入力画像を分割して得られたウィンドウから 1 つのウィンドウを  $W^{\xi, \zeta}(x, y)$  を選択する。選択の順序に関しては特に限定は無く、今まで選択されていないウィンドウを 1 つ選択すればよい。

【0053】

次いで、ウィンドウ選択処理（ステップ S 3 2）で選択されたウィンドウ  $W^{\xi, \zeta}(x, y)$  に対し、基底組化処理を行う（ステップ S 3 3）。ここでは、選択されたウィンドウ  $W^{\xi, \zeta}(x, y)$  を複数の離散的な 1 次元適応化基底組  $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$  に変換する。この基底組化での処理は、第 2 の実施形態で説明した、複数の 1 次元適応化基底組に符号化する方法と同様の方法を用いればよい。ここで設定する 1 次元適応化基底組  $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$  の定義域は、基本的には  $0 \leq x < L_x$ 、 $0 \leq y < L_y$  とする。また、上記説明した  $x + \xi L_x \geq x_{\max}$ 、若しくは、 $y + \zeta L_y \geq y_{\max}$  となるウィンドウでは、そのウィンドウのサイズを

【数 16】

$$\tilde{L}_x, \tilde{L}_y$$

としたとき、

【数 17】

$$0 \leq x < \tilde{L}_x$$

とすればよい。

【0054】

基底組化処理（ステップ S 3 3）では、第 2 の実施形態と同様のものであるので、詳細な説明を割愛する。この基底組化処理（ステップ S 3 3）により、ウィンドウ選択（ステップ S 3 2）で選択された 1 つのウィンドウ  $W^{\xi, \zeta}(x, y)$  の 1 次元適応化基底組変換が完了したら、次の選択終了判定処理（ステップ S 3 4）に進む。

【0055】

選択終了判定処理（ステップ S 3 4）では、全てのウィンドウを選択したかどうかを判定して、まだ選択されていないウィンドウがある場合は、ウィンドウ選択（ステップ S 3 2）に戻り、未選択のウィンドウを選択して、そのウィンドウに対して上記基底組化（ステップ S 3 3）により、複数の 1 次元適応化基底組への変換を行う。全てのウィンドウを選択した場合は、次の DCT 変換処理（ステップ S 3 5）に進む。

【0056】

DCT 変換処理（ステップ S 3 5）では、基底組化処理（ステップ S 3 3）で生成された、各ウィンドウ  $W^{\xi, \zeta}(x, y)$  の、複数の離散的な 1 次元適応化基底組  $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$  のそれぞれを、DCT ベクトル  $U^{\xi, \zeta}_1(x)$ 、 $V^{\xi, \zeta}_1(y)$  に離散コサイン変換する。各 1 次元適応化基底に対する離散コサイン変換は、以下の変換式を用いて行えばよい。



【数 1 8】

$$U^{\xi, \xi_1}(x) = \sum_{i=0}^{L_x-1} \left[ \sqrt{\frac{2}{L_x}} \cdot c_x \cdot \cos \left\{ \frac{(x-1)(2i-1)}{2L_x} \pi \right\} \cdot X^{\xi, \xi_1}(i) \right]$$

$$V^{\xi, \xi_1}(y) = \sum_{i=0}^{L_y-1} \left[ \sqrt{\frac{2}{L_y}} \cdot c_y \cdot \cos \left\{ \frac{(y-1)(2i-1)}{2L_y} \pi \right\} \cdot Y^{\xi, \xi_1}(i) \right]$$

ここで  $c_x$  は、 $x=0$  のとき、 $c_x=1/\sqrt{2}$  である。また、 $x \neq 0$  のとき、 $c_x=1$  である。 $c_y$  も同様で、 $y=0$  のとき、 $c_y=1/\sqrt{2}$  であり、 $y \neq 0$  のとき、 $c_y=1$  である。

【0 0 5 7】

上記変換により、基底組化処理（ステップ S 3 3）で生成され、全ての離散的な 1 次元適応化基底  $X^{\xi, \xi_1}(x)$ 、 $Y^{\xi, \xi_1}(y)$  それぞれを、DCT ベクトル  $U^{\xi, \xi_1}(x)$ 、 $V^{\xi, \xi_1}(y)$  に変換する。

【0 0 5 8】

次いで、量子化処理（ステップ S 3 6）において、DCT 変換処理（ステップ S 3 5）により得られた全ての DCT ベクトル  $U^{\xi, \xi_1}(x)$ 、 $V^{\xi, \xi_1}(y)$  に対して量子化を行い、量子化基底  $U_q^{\xi, \xi_1}(x)$ 、 $V_q^{\xi, \xi_1}(y)$  を生成する。量子化は、DCT ベクトルの要素数の等しい 1 次元の量子化係数ベクトル  $Q_x(x)$  及び  $Q_y(y)$  を用いて以下に示すように行う。

【数 1 9】

$$U_q^{\xi, \xi_1}(x) = \left\lfloor \frac{U^{\xi, \xi_1}(x)}{Q_x(x)} \right\rfloor$$

$$V_q^{\xi, \xi_1}(y) = \left\lfloor \frac{V^{\xi, \xi_1}(y)}{Q_y(y)} \right\rfloor$$

【0 0 5 9】

ここで、

[a]

は、 $a$  を超えない最大の整数を意味し、例えば、

[1.5]=1

となる。

【0 0 6 0】

この量子化係数ベクトル  $Q_x(x)$  及び  $Q_y(y)$  は、J P E G 形式の量子化係数と同様、高周波成分、つまり  $x$  が大きい成分ほど量子化係数を大きくするのが好ましい。また、量子化係数ベクトル  $Q_x(x)$  及び  $Q_y(y)$  の値は、DCT ベクトルのレンジと、各周波数成分で求められる分解能に応じて予め定めておけばよい。

【0 0 6 1】

最後に、量子化処理（ステップ S 3 6）により得られた量子化基底  $U_q^{\xi, \xi_1}(x)$ 、 $V_q^{\xi, \xi_1}(y)$  をエントロピー符号化する（ステップ S 3 7）。このエントロピー符号化処理では、全ての量子化基底  $U_q^{\xi, \xi_1}(x)$ 、 $V_q^{\xi, \xi_1}(y)$  を符号化データとして符号化を行う。ここでのエントロピー符号化の方法としては、例えばハフマン符号化や、算術符号化等の公知の符号化方法を用いればよい。そして、エントロピー符号化された量子化基底と、ウィンドウごとの基底数や、ウィンドウサイズ等の復号化に必要なインデックス情報を、符号化データとして記録して処理を終了する。インデックス情報に関しては、各インデックス情報を、予め定めて規格化されたものを用いれば省略することも可能で

ある。

#### 【0062】

本第3の実施形態では、DCT変換処理（ステップS35）において、複数の離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ に対し、そのまま離散コサイン変換を行ったが、特にこれに限るものではなく、各1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ を分割し、それぞれに対して離散コサイン変換し、量子化して、エントロピー符号化するようにしても構わない。

#### 【0063】

上記方法により符号化した符号化データを復号化画像 $f_{\text{decode}}(x, y)$ に復号するのは、適宜インデックス情報を用いながら、まずエントロピー復号化により量子化基底 $U_q^{\xi, \zeta}_1(x)$ 、 $V_q^{\xi, \zeta}_1(y)$ を生成して逆量子化し、復号化DCTベクトル $U_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $V_{\text{decode}}^{\xi, \zeta}_1(y)$ を復元して、それに対して離散コサイン変換の逆変換を行うことにより、復号化1次元適応化基底 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ を復元する。次いで、復元した復号化1次元適応化基底 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ に基づき、復号化ウィンドウ $W_{\text{decode}}^{\xi, \zeta}(x, y)$ を下記に示す式により生成する。

#### 【数20】

$$W_{\text{decode}}^{\xi, \zeta}(x, y) = \sum_{n=0}^{k^{\xi, \zeta}-1} [X_{\text{decode}}^{\xi, \zeta}_n(x) \cdot Y_{\text{decode}}^{\xi, \zeta}_n(y)]$$

#### 【0064】

上式の $k^{\xi, \zeta}$ は、ウィンドウごとの、選択され修正された基底組数である。復号化画像 $f_{\text{decode}}(x, y)$ は、上記生成した復号ウィンドウ $W_{\text{decode}}^{\xi, \zeta}(x, y)$ を合成することで得ることができる。

#### 【0065】

本第3の実施形態では、第2の実施形態と同様、生成された離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ から要素復元画像を復元して、差分被符号化画像を生成する。量子化のための誤差を吸収するため、順次生成される離散的な1次元適応化基底組 $X^{\xi, \zeta}_1(x)$ 、 $Y^{\xi, \zeta}_1(y)$ に対し、1次元適応化基底組が生成されるごとにDCT変換（ステップS35）、及び量子化（ステップS36）を行い、量子化後の量子化基底 $U_q^{\xi, \zeta}_1(x)$ 、 $V_q^{\xi, \zeta}_1(y)$ から、ただちに上記復号化と同様の手法で、逆量子化、離散コサイン変換の逆変換を行い、復号化1次元基底 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ を生成し、この復号化1次元適応化基底を用いて、要素復元画像を復元して、差分被符号化画像を生成するようにしてもよい。

#### 【0066】

上記第3の実施形態に示した符号化方法では、入力画像を複数のウィンドウに分割して基底組化することにより、ウィンドウに対応する基底組数をウィンドウごとに可変にすることができる。そのため、濃度分布が単調なウィンドウ、例えば図5に示したような濃度分布であるウィンドウならば、そのウィンドウは少ない基底組数で誤差の少ない符号化をすることができ、ウィンドウが自然画像のような濃度分布であれば多くの基底組を用いて符号化するというように、基底組数の最適化を行うことができる。また、基底組化したものを離散コサイン変換して量子化し、それをエントロピー符号化することで、さらに高効率の符号化が可能になる。

#### 【0067】

以上、第3の実施の形態では、有限範囲で定義される離散的な2次元濃淡画像 $f(x, y)$ を、複数の離散的な1次元適応化基底組 $X_{\text{decode}}^{\xi, \zeta}_1(x)$ 、 $Y_{\text{decode}}^{\xi, \zeta}_1(y)$ に変換し、それを符号化する方法の例として説明した。

#### 【0068】

<第4の実施の形態>

本第4の実施の形態では、第3の実施形態の変形として、有限範囲で定義される離散的な2次元濃淡画像  $f(x, y)$  を2次元離散コサイン変換して量子化し、量子化周波数画像  $f^{u,v}_1(\xi, \zeta)$  を生成して、それを複数の離散的な1次元適応化基底組  $X^{u,v}_1(x)$ 、 $Y^{u,v}_1(y)$  に符号化する方法を説明する。

#### 【0069】

装置構成は第1の実施形態と同様であり、以下、本第4の実施形態の処理を図4のフローチャートに従って説明する。

#### 【0070】

まず、画像入力(ステップS40)において、符号化される離散的な2次元濃淡画像である入力画像  $f(x, y)$  を入力する。ここでも  $x, y$  は、上記第2の実施形態と同様の有限の定義域をもち、整数であるとする。

#### 【0071】

次に、2次元DCT変換処理を行う(ステップS41)。ここでは、入力画像  $f(x, y)$  を所定サイズ、例えば縦幅8、横幅8の画素サイズの分割画像  $W^{\xi, \zeta}(x, y)$  に分割する。ここでの分割画像  $W^{\xi, \zeta}(x, y)$  は、上記第3の実施形態で示したウィンドウ  $W^{\xi, \zeta}(x, y)$  のものである。ステップS41では、それぞれの分割画像  $W^{\xi, \zeta}(x, y)$  に対して2次元離散コサイン変換を行う。

#### 【0072】

次いで、2次元離散コサイン変換されたデータを量子化する(ステップS42)。この量子化処理では、所定の量子化係数  $Q(x, y)$  を用いて量子化し、量子化画像  $W_q^{\xi, \zeta}(x, y)$  を生成する。つまり、JPEG形式の符号化における、2次元離散コサイン変換後、量子化というステップと同様の変換を行う。ここでの変換はJPEG形式の変換と同様であるので詳細な説明は割愛する。

#### 【0073】

本第4の実施形態では、入力画像  $f(x, y)$  を縦幅8、横幅8のサイズの分割画像  $W^{\xi, \zeta}(x, y)$  に分割して2次元離散コサイン変換を行い、高周波で値が大きくなるような量子化係数  $Q(x, y)$  を用いて量子化するが、特にこれに限定するものではなく、その他のサイズで分割する、若しくは他の量子化係数を用いて量子化しても構わない。

#### 【0074】

次いで、周波数選択処理を行う(ステップS43)。ここでは、周波数パラメータ  $u, v$  を選択し、周波数画像  $f^{u,v}(\xi, \zeta)$  を生成する。ここで、周波数パラメータ  $u, v$  は0以上の整数であり、本第4の実施形態では2次元DCT変換(ステップS41)で、入力画像  $f(x, y)$  を縦幅8、横幅8のサイズの分割画像  $W^{\xi, \zeta}(x, y)$  に分割したので、7までの整数である。そのためここでは、周波数パラメータ  $u, v$  は、ともにそれぞれ0から7の8種類あるので、計  $8 \times 8 = 64$  の組み合わせから、1つの周波数パラメータの組み合わせを選択する。ここで生成する周波数画像  $f^{u,v}(\xi, \zeta)$  は、量子化(ステップS42)において生成された量子化画像  $W_q^{\xi, \zeta}(x, y)$  を用いて以下のように表わされる。

$$f^{u,v}(\xi, \zeta) = W_q^{\xi, \zeta}(u, v)$$

上式に示すとおり、周波数画像  $f^{u,v}(\xi, \zeta)$  は、各量子化画像  $W_q^{\xi, \zeta}(x, y)$  中の、選択された周波数  $u, v$  の成分を並べたものである。

#### 【0075】

次の基底組化処理(ステップS44)では、選択された周波数パラメータ  $u, v$  に対応して生成された周波数画像  $f^{u,v}(\xi, \zeta)$  を、複数の離散的な1次元適応化基底組  $X^{u,v}_1(\xi)$ 、 $Y^{u,v}_1(\zeta)$  に変換する。ここでの処理は第2の実施形態で示した、複数の1次元適応化基底組に符号化する処理と同様であるので説明を割愛する。

#### 【0076】

選択終了判定(ステップS45)では、全ての周波数パラメータ  $u, v$  を選択したかどうかを判定する。本第4の実施形態では、周波数パラメータ  $u, v$  の組み合わせは、全64通りであるので、それを全て選択して、基底組化(ステップS44)によりそれぞれの

周波数パラメータ  $u$ ,  $v$  に対応する周波数画像  $f^{u,v}(\xi, \zeta)$  の基底組化が完了していれば、次のエントロピー符号化（ステップ S 4 6）に進む。64 通り全ての周波数パラメータ  $u$ ,  $v$  の組み合わせを選択していない場合は、周波数選択（ステップ S 4 3）に戻り、まだ選択されていない周波数パラメータ  $u$ ,  $v$  の組み合わせを選択し、基底組化（ステップ S 4 4）により、選択された周波数パラメータ  $u$ ,  $v$  に対応する周波数画像  $f^{u,v}(\xi, \zeta)$  の基底組化を行う。

#### 【0077】

エントロピー符号化（ステップ S 4 6）では、基底組化（ステップ S 4 4）において生成された基底組  $X^{u,v}_1(\xi)$ 、 $Y^{u,v}_1(\zeta)$  をエントロピー符号化し、符号化データとして、復元に必要なインデックス情報とともに記録して処理を終了する。ここでの処理は、上記第 3 の実施形態で説明したものと同様の処理を行えばよい。

#### 【0078】

この符号化データから、復号化画像を生成する方法は、第 3 の実施形態と同様に、適宜インデックス情報を用いながら、まずエントロピー復号化により 1 次元基底組を復元して、その基底組を用いて周波数画像を復元する。この周波数画像から量子化画像を生成して逆量子化し、それに対し 2 次元離散コサイン変換の逆変換を行うことで復号化画像を復元することができる。

#### 【0079】

第 3 の実施形態が入力画像を空間的に分割して符号化したのに対し、上記第 4 の実施形態では、周波数空間で分割して符号化する方法を示した。本第 4 の実施形態では、周波数空間で分割したものに対して 1 次元の基底組化を行ったが、周波数空間で分割したものをさらに空間的に分割して基底組化を行っても構わない。また、エントロピー符号化（ステップ S 4 6）では、生成された基底組をそのままエントロピー符号化したが、1 次元適応化基底組を量子化してからエントロピー符号化しても良いし、実施例 3 と同様に、もう 1 度離散コサイン変換（1 次元）し、量子化してエントロピー符号化しても構わない。

#### 【0080】

以上、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を 2 次元離散コサイン変換して量子化し、周波数画像  $f^{u,v}(\xi, \zeta)$  を生成して、それを複数の離散的な 1 次元適応化基底組  $X^{u,v}_1(\xi)$ 、 $Y^{u,v}_1(\zeta)$  に符号化する方法の例として説明した。

#### 【0081】

##### <第 5 の実施形態>

上記第 1 乃至第 4 の実施形態では、全て 2 次元データに対する符号化について説明したが、本発明の符号化方法はこれに限るものではなく、多次元のデータに関しても適用できる。そこで本第 5 の実施形態では、第 2 の実施形態の変形として、有限範囲で定義される離散的な 3 次元データ  $f(x, y, z)$  を、複数の離散的な 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$ 、 $Z_1(z)$  に符号化する方法を説明する。

#### 【0082】

符号化される離散的な 3 次元データとしては、3 次元空間内の離散的な密度分布データや、離散的な 2 次元濃淡画像の離散的な時系列データが考えられる。ここでは、3 次元データの各要素（2 次元濃淡画像の離散的な時系列データの場合には、所定時刻の 2 次元濃淡画像の画素）が 1 バイトで表現されるものとして説明する。

#### 【0083】

まずはこのような符号化される 3 次元データ  $f(x, y, z)$  を入力する。ここでも第 2 の実施形態と同様に、 $x$ 、 $y$ 、 $z$  は所定の定義域  $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$ 、 $0 \leq z < z_{\max}$  を有し、整数であるとする。この 3 次元データに対し、下記の誤差  $E$ 、修正量  $\Delta X_1(x)$ 、 $\Delta Y_1(y)$ 、 $\Delta Z_1(z)$  に基づいて、第 2 の実施形態と同様に 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$ 、 $Z_1(z)$  を修正していくことで符号化を行う。

【数 2 1】

$$E = \sum_x \sum_y \sum_z \left[ \frac{1}{2} \{f_i(x, y, z) - \tilde{f}_i(x, y, z)\}^2 \right]$$

$$\Delta X_i(x) = -\eta_x \cdot \frac{\partial E}{\partial X_i(x)} = \eta_x \cdot \sum_y \sum_z [\{f_i(x, y, z) - \tilde{f}_i(x, y, z)\} \cdot Y_i(y) \cdot Z_i(z)]$$

$$\Delta Y_i(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_i(y)} = \eta_y \cdot \sum_x \sum_z [\{f_i(x, y, z) - \tilde{f}_i(x, y, z)\} \cdot X_i(x) \cdot Z_i(z)]$$

$$\Delta Z_i(z) = -\eta_z \cdot \frac{\partial E}{\partial Z_i(z)} = \eta_z \cdot \sum_x \sum_y [\{f_i(x, y, z) - \tilde{f}_i(x, y, z)\} \cdot X_i(x) \cdot Y_i(y)]$$

上式中の  $f_i(x, y, z)$  や、 $\tilde{f}_i(x, y, z)$  は、第 2 の実施形態での差分被符号化画像  $f_i(x, y)$ 、要素復元画像  $\tilde{f}_i(x, y)$  と同意であり、次元を 3 次元としただけのものである。その他のパラメータ等に関しても、第 2 の実施形態と同様のものであるので説明は割愛する。

【0084】

上記誤差  $E$ 、修正量  $\Delta X_i(x)$ 、 $\Delta Y_i(y)$ 、 $\Delta Z_i(z)$  に基づき、第 2 の実施形態で説明したのと同様の方法で、1 次元適応化基底組に対して修正をしていくことで、3 次元データを 1 次元適応化基底組  $X_i(x)$ 、 $Y_i(y)$ 、 $Z_i(z)$  に符号化する。各基底の修正に関しては、選択された 1 次元適応化基底組  $X_i(x)$ 、 $Y_i(y)$ 、 $Z_i(z)$  に対して、修正後の各 1 次元適応化基底組を  $X'_i(x)$ 、 $Y'_i(y)$ 、 $Z'_i(z)$  としたとき、次のように修正する。

【数 2 2】

$$X'_i(x) = X_i(x) + \Delta X_i(x)$$

$$Y'_i(y) = Y_i(y) + \Delta Y_i(y)$$

$$Z'_i(z) = Z_i(z) + \Delta Z_i(z)$$

【0085】

最後に、誤差  $E$  が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した 1 次元適応化基底組  $X_i(x)$ 、 $Y_i(y)$ 、 $Z_i(z)$  を記録して処理を終了する。ここでの処理に関しても、第 2 の実施形態と同様であるので、詳細な説明は割愛する。この場合、1 次元適応化基底組  $X_i(x)$ 、 $Y_i(y)$ 、 $Z_i(z)$  から復元される復号化データ  $f_{\text{decode}}(x, y, z)$  は以下ようになる。

【数 2 3】

$$f_{\text{decode}}(x, y, z) = \sum_{n=0}^{k-1} [X_n(x) \cdot Y_n(y) \cdot Z_n(z)]$$

上式中、 $k$  は基底組数である。このように符号化されるデータの次元数が増えた場合でも、上記のように誤差と修正量を求めることで符号化が可能になる。本実施形態では、入力された 3 次元データをそのまま 1 次元適応化基底に符号化しているが、上記第 3 の実施形態等で示したように、所定範囲で分割したり、離散コサイン変換等により周波数空間で分割したりしても構わない。

【0086】

以上、有限範囲で定義される離散的な 3 次元データ  $f(x, y, z)$  を、複数の離散的な 1 次元適応化基底組  $X_i(x)$ 、 $Y_i(y)$ 、 $Z_i(z)$  に符号化する方法の例として説明した。

【0087】

&lt;第6の実施形態&gt;

上記第1乃至第5の実施形態では、誤差評価関数として、各要素の差の2乗誤差を用いたが、本発明の符号化方法はこれに限るものではなく、その他の誤差評価関数を用いても適用可能である。そこで本発明の第6の実施形態では、第2の実施形態の変形として、aを参照データ、つまり符号化されるデータとし、bを評価データ、つまり符号化されたデータから復元されるデータとした時に、誤差評価関数として  $\sum \{1/2 \times (a^2 - ab)^2\}$  を用い、有限範囲で定義される離散的な2次元濃淡画像  $f(x, y)$  を、複数の離散的な1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に符号化する方法を説明する。

【0088】

まず、符号化される離散的な2次元濃淡画像  $f(x, y)$  を入力する。ここでも第2の実施形態と同様に、 $x, y$  は所定の定義域  $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$  を有し、整数であるとする。この2次元濃淡画像  $f(x, y)$  に対し、下記の誤差  $E$ 、修正量  $\Delta X_1(x)$ 、 $\Delta Y_1(y)$  に基づいて、第2の実施形態と同様に1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  を修正していくことで符号化を行う。

【数24】

$$E = \sum_x \sum_y \left[ \frac{1}{2} \{f_i(x, y)^2 - f_i(x, y) \cdot \tilde{f}_i(x, y)\}^2 \right]$$

$$\Delta X_i(x) = -\eta_x \cdot \frac{\partial E}{\partial X_i(x)} = \eta_x \cdot \sum_y [f_i(x, y) \cdot \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot Y_i(y)]$$

$$\Delta Y_i(y) = -\eta_y \cdot \frac{\partial E}{\partial Y_i(y)} = \eta_y \cdot \sum_x [f_i(x, y) \cdot \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot X_i(x)]$$

【0089】

上式中の  $f_i(x, y)$  や、 $\tilde{f}_i(x, y)$  は、第2の実施形態での差分被符号化画像  $f_1(x, y)$ 、要素復元画像  $\tilde{f}_1(x, y)$  と同様のものである。また、その他のパラメータ等に関しても、第2の実施形態と同様のものであるので説明は割愛する。

【0090】

上記誤差  $E$ 、修正量  $\Delta X_1(x)$ 、 $\Delta Y_1(y)$  に基づき、第2の実施形態で説明したのと同様の方法で、1次元適応化基底組を修正をしていくことで、1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に符号化する。各1次元適応化基底組の修正に関しては、選択された1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に対して、修正後の各1次元適応化基底組を  $X'_1(x)$ 、 $Y'_1(y)$ 、修正前の基底を  $X_1(x)$ 、 $Y_1(y)$  とした時に、次のように修正する。

【数25】

$$X'_1(x) = X_1(x) + \Delta X_1(x)$$

$$Y'_1(y) = Y_1(y) + \Delta Y_1(y)$$

【0091】

最後に、誤差  $E$  が所定値以下になるか、若しくは基底組数が所定数に達した場合に、今までに選択して修正した1次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  を記録して処理を終了する。ここでの処理に関しても、第2の実施形態と同様であるので、詳細な説明は割愛する。この場合の復号化画像  $f_{\text{decode}}(x, y)$  は、第2の実施形態で示したものと同様である。

【0092】

このように、2乗誤差以外の誤差評価関数を用いても、上記のように誤差と修正量を求めることで符号化が可能になる。本第6の実施形態では、誤差評価関数に相互相関関数を

用いたが、本発明の符号化方法はこれに限るものではなく、被符号化データに近づくことにより誤差が減少するようなものであれば、その他の誤差評価関数であっても構わない。また、そのような誤差評価関数のうち、上記示したような、修正すべきパラメータ、ここでは  $X_1(x)$ 、 $Y_1(y)$  での偏微分が解析的に求められることが好ましいが、解析的に偏微分を求められない場合は、修正すべきパラメータに微小変化を与えた時の誤差の増減から修正量を決め手やるようにすればよい。

#### 【0093】

以上、誤差評価関数に  $\Sigma 1/2 \times \{a^2 - ab\}^2$  を用い、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数の離散的な 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に符号化する方法の例として説明した。なお、本第 6 の実施形態では、特に第 2 の実施形態に適用する例を説明したが、他の実施形態にも同様に適用できるのは明らかである。

#### 【0094】

##### <第 7 の実施形態>

上記第 1 乃至第 6 の実施形態では、多次元のデータを複数の 1 次元適応化基底組に符号化する方法を示したが、本発明の符号化方法はこれに限るものではなく、その他のデータに符号化する方法においても適用可能である。そこで本発明の第 7 の実施形態では、第 2 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数の離散的な 2 次元適応化基底  $S_1(x_s, y_s)$  に符号化する方法を説明することとする。

#### 【0095】

まず、符号化される離散的な 2 次元濃淡画像  $f(x, y)$  を入力する。ここでも第 2 の実施形態と同様に、 $x, y$  は所定の定義域  $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$  を有し、整数であるとする。次に基底として、離散的な 2 次元適応化基底  $S_1(x_s, y_s)$  を 1 つ選択する。ここで  $x_s, y_s$  は、 $x_{\max}$  もしくは  $y_{\max}$  が偶数である場合は、それぞれの定義域が  $0 \leq x_s < (x_{\max}/2) + 1$ 、 $0 \leq y_s < (y_{\max}/2) + 1$  となる整数である。 $x_{\max}$ 、 $y_{\max}$  が奇数である場合は、それぞれの定義域は  $0 \leq x_s < (x_{\max}/2) + 0.5$ 、 $0 \leq y_s < (y_{\max}/2) + 0.5$  となる整数である。また、添字  $l$  は基底番号であり、特に限定はしないが、本第 7 の実施形態でも 0 から順に、予め定めた基底数まで順に選択していくことにする。

#### 【0096】

次に、選択した基底番号  $l$  の 2 次元適応化基底  $S_l(x_s, y_s)$  から、要素復元画像  $f_l(x, y)$  を次のように算出する。

## 【数 2 6】

i)  $x, y$ : 偶数

$$\tilde{f}_l(x, y) = S_l\left(\frac{x}{2}, \frac{y}{2}\right)$$

ii)  $x, y$ : 奇数

$$\tilde{f}_l(x, y) = \frac{1}{4} \left\{ S_l\left(\frac{x-1}{2}, \frac{y-1}{2}\right) + S_l\left(\frac{x+1}{2}, \frac{y-1}{2}\right) + S_l\left(\frac{x-1}{2}, \frac{y+1}{2}\right) + S_l\left(\frac{x+1}{2}, \frac{y+1}{2}\right) \right\}$$

iii)  $x$ : 偶数、 $y$ : 奇数

$$\tilde{f}_l(x, y) = \frac{1}{2} \left\{ S_l\left(\frac{x}{2}, \frac{y-1}{2}\right) + S_l\left(\frac{x}{2}, \frac{y+1}{2}\right) \right\}$$

iv)  $x$ : 奇数、 $y$ : 偶数

$$\tilde{f}_l(x, y) = \frac{1}{2} \left\{ S_l\left(\frac{x-1}{2}, \frac{y}{2}\right) + S_l\left(\frac{x+1}{2}, \frac{y}{2}\right) \right\}$$

## 【0097】

上記求めた要素復元画像  $f^{-1}(x, y)$  を用いて、誤差  $E$ 、及び修正量  $\Delta S_l(x_s, y_s)$  を以下のように算出する。

## 【数 2 7】

$$E = \sum_x \sum_y \left[ \frac{1}{2} \{ f_l(x, y) - \tilde{f}_l(x, y) \}^2 \right]$$

$$\Delta S_l(x_s, y_s) = -\eta \cdot \frac{\partial E}{\partial S_l(x_s, y_s)} = \eta \cdot \sum_{n=-1}^1 \sum_{m=-1}^1 \left[ \{ f_l(2x_s + n, 2y_s + m) - \tilde{f}_l(2x_s + n, 2y_s + m) \} C(m, n) \right]$$

## 【0098】

上式の  $f_l(x, y)$  は差分被符号化画像であり、第2の実施形態と同様のものである。また修正量  $\Delta S_l(x_s, y_s)$  を算出する式において  $C(m, n)$  は次の値を有する。

・  $n=0$ 、且つ、 $m=0$  の場合

$$C(n, m) = 1$$

・  $n=\pm 1$ 、且つ、 $m=0$  の場合、又は、 $n=0$ 、且つ、 $m=\pm 1$  の場合

$$C(n, m) = 0.5$$

・  $n=\pm 1$ 、且つ、 $m=\pm 1$  の場合

$$C(n, m) = 0.25$$

ただし、上記算出式において、 $2x_s + n < 0$ 、 $2x_s + n \geq x_{\max}$ 、 $2y_s + m < 0$ 、又は、 $2y_s + m \geq y_{\max}$  のいずれかに場合には、 $C(n, m) = 0$  とする。

## 【0099】

また  $\eta$  は修正パラメータであり、第2の実施形態で説明した  $\eta_x$ 、 $\eta_y$  と同様のものである。上記誤差  $E$ 、修正量  $\Delta S_l(x_s, y_s)$  に基づき、第2の実施形態で説明したのと同様の方法で、2次元適応化基底に対して修正をしていくことで、入力画像  $f(x, y)$  を2次元適応化基底  $S_l(x_s, y_s)$  に符号化する。

## 【0100】

基底の修正に関しては、選択された2次元適応化基底  $S_l(x_s, y_s)$  を次のように修正する。式中、 $S'_l(x_s, y_s)$  が修正後、 $S_l(x_s, y_s)$  が修正前である。

$$S'_l(x_s, y_s) = S_l(x_s, y_s) + \Delta S_l(x_s, y_s)$$



最後に、誤差  $E$  が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した 2 次元適応化基底  $S_1(x_s, y_s)$  を記録して処理を終了する。ここの処理に関しても、第 2 の実施形態と同様であるので、詳細な説明は割愛する。この方法により符号化した符号化データを復号化画像  $f_{\text{decode}}(x, y)$  に復号するのは、下記に示すように行えばよい。

【数 28】

i)  $x, y$  : 偶数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} S_n\left(\frac{x}{2}, \frac{y}{2}\right)$$

ii)  $x, y$  : 奇数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[ \frac{1}{4} \left\{ S_n\left(\frac{x-1}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x-1}{2}, \frac{y+1}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y+1}{2}\right) \right\} \right]$$

iii)  $x$  : 偶数、 $y$  : 奇数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[ \frac{1}{2} \left\{ S_n\left(\frac{x}{2}, \frac{y-1}{2}\right) + S_n\left(\frac{x}{2}, \frac{y+1}{2}\right) \right\} \right]$$

iv)  $x$  : 奇数、 $y$  : 偶数

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[ \frac{1}{2} \left\{ S_n\left(\frac{x-1}{2}, \frac{y}{2}\right) + S_n\left(\frac{x+1}{2}, \frac{y}{2}\right) \right\} \right]$$

【0101】

上式の  $k$  は選択され修正された基底数である。本第 7 の実施形態では、生成する 2 次元適応化基底  $S_1(x_s, y_s)$  を、要素数が入力画像  $f(x, y)$  の約  $1/4$  の画像としたが、これに限るものではなく、これとは異なる要素数の 2 次元適応化基底を用いて符号化しても構わない。

【0102】

以上、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数の離散的な 2 次元適応化基底  $S_1(x_s, y_s)$  に符号化する第 7 の実施形態を説明した。

【0103】

< 第 8 の実施形態 >

上記第 1 乃至第 7 の実施形態では、適応化基底として 1 次元適応化基底や 2 次元適応化基底を用いて符号化する方法を示したが、本発明の符号化方法はこれに限るものではなく、いくつかのパラメータで構成されるような適応化基底を用いるような場合でも適用可能である。そこで本発明の第 8 の実施形態では、第 2 の実施形態の変形として、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数のパラメータで構成される適応化基底に符号化する例を説明する。以下では、具体的な例として、4 つのパラメータ  $A_1$ 、 $p_1$ 、 $q_1$ 、 $\rho_1$  から構成される適応化基底  $G_1(A_1, p_1, q_1, \rho_1)$  に符号化する方法を説明する。

【0104】

まず、符号化される離散的な 2 次元濃淡画像  $f(x, y)$  を入力する。ここでも第 2 の実施形態と同様に、 $x, y$  は所定の定義域  $0 \leq x < x_{\max}$ 、 $0 \leq y < y_{\max}$  を有し、整数であるとする。次に初期の適応化基底として、適応化基底  $G_1$  を 1 つ選択する。添字 1 は基底番号であり、特に限定はしないが、本第 8 の実施形態でも 0 から順に、予め定めた基底数まで順に選択していくことにする。

【0105】

次に、選択した基底番号 1 の適応化基底  $G_1$  から、要素復元画像  $f^{-1}(x, y)$  を次のように算出する。

【数 29】

$$\tilde{f}_i(x, y) = A_i \cdot \exp\left\{\frac{(x-p_i)^2 + (y-q_i)^2}{2\sigma_i^2}\right\}$$

【0106】

次に、上記求めた要素復元画像  $f^{-1}(x, y)$  を用いて、誤差  $E$ 、及び修正量  $\Delta A_i$ 、 $\Delta p_i$ 、 $\Delta q_i$ 、 $\Delta \rho_i$  を以下のように算出する。

【数 30】

$$E = \sum_x \sum_y \left[ \frac{1}{2} \{f_i(x, y) - \tilde{f}_i(x, y)\}^2 \right]$$

$$\Delta A_i = -\eta_A \cdot \frac{\partial E}{\partial A_i} = \eta_A \cdot \sum_x \sum_y \left[ \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \frac{\tilde{f}_i(x, y)}{A_i} \right]$$

$$\Delta p_i = -\eta_p \cdot \frac{\partial E}{\partial p_i} = -\eta_p \cdot \sum_x \sum_y \left[ \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{x-p_i}{\sigma_i^2} \right]$$

$$\Delta q_i = -\eta_q \cdot \frac{\partial E}{\partial q_i} = -\eta_q \cdot \sum_x \sum_y \left[ \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{y-q_i}{\sigma_i^2} \right]$$

$$\Delta \sigma_i = -\eta_\sigma \cdot \frac{\partial E}{\partial \sigma_i} = -\eta_\sigma \cdot \sum_x \sum_y \left[ \{f_i(x, y) - \tilde{f}_i(x, y)\} \cdot \tilde{f}_i(x, y) \cdot \frac{(x-p_i)^2 + (y-q_i)^2}{\sigma_i^3} \right]$$

【0107】

上式の  $f_i(x, y)$  は差分被符号化画像であり、第 2 の実施形態と同様のものである。また  $\eta_A$ 、 $\eta_p$ 、 $\eta_q$ 、 $\eta_\sigma$  は修正パラメータであり、第 2 の実施形態で説明した  $\eta_x$ 、 $\eta_y$  と同様のものである。上記誤差  $E$ 、修正量  $\Delta A_i$ 、 $\Delta p_i$ 、 $\Delta q_i$ 、 $\Delta \rho_i$  に基づき、第 2 の実施形態で説明したのと同様の方法で、適応化基底を構成する各パラメータに対して修正をしていくことで、適応化基底  $G_1$  に符号化する。

【0108】

適応化基底の修正に関しては、選択された基底  $G_1$  を構成する各パラメータ  $A_i$ 、 $p_i$ 、 $q_i$ 、 $\rho_i$  を以下のようにして修正する。 $A'_i$ 、 $p'_i$ 、 $q'_i$ 、 $\rho'_i$  が修正後、 $A_i$ 、 $p_i$ 、 $q_i$ 、 $\rho_i$  が修正前である。

$$A'_i = A_i + \Delta A_i$$

$$p'_i = p_i + \Delta p_i$$

$$q'_i = q_i + \Delta q_i$$

$$\rho'_i = \rho_i + \Delta \rho_i$$

【0109】

最後に、誤差  $E$  が所定値以下になるか、若しくは基底数が所定数に達した場合に、今までに選択して修正した適応化基底  $G_1$  を構成する各パラメータ  $A_i$ 、 $p_i$ 、 $q_i$ 、 $\rho_i$  を記録出力して処理を終了する。ここでの処理に関しても、第 2 の実施形態と同様であるので、詳細な説明は割愛する。この方法により符号化した符号化データを復号化画像  $f_{\text{decode}}(x, y)$  に復号するのは、下記に示すように行えばよい。

## 【数 3 1】

$$f_{\text{decode}}(x, y) = \sum_{n=0}^{k-1} \left[ A_n \cdot \exp \left\{ \frac{(x-p_n)^2 + (y-q_n)^2}{2\sigma_n^2} \right\} \right]$$

上式の  $k$  は選択され修正された基底数である。本第 8 の実施形態では、4 つのパラメータにより構成される適応化基底  $G_1$  を用いたが、これに限るものではなく、これとは異なるパラメータ数の適応化基底を用いて符号化しても構わない。

## 【0 1 1 0】

以上、有限範囲で定義される離散的な 2 次元濃淡画像  $f(x, y)$  を、複数のパラメータから構成される適応化基底  $G_1$  に符号化する方法の例として説明した。

## 【0 1 1 1】

## &lt; 第 9 の実施形態 &gt;

本第 9 の実施形態では、所定の 2 次元荷重分布データによる、2 次元分布データに対する離散的コンボリューション演算において、2 次元荷重分布データを第 2 の実施形態のデータ符号化処理により 1 次元適応化基底組に変換し、それを用いて 2 次元分布データに対する離散的コンボリューション演算を行う方法を説明する。ここで示す離散的コンボリューション演算は、例えば、Convolutional Neural Networks (LeCun, Y. and Bengio, Y., 1995, "Convolutional Networks for Images Speech, and Time Series" in Handbook of Brain Theory and Neural Networks (M. Arbib, Ed.), MIT Press, pp. 255-258) における 1 つの特徴検出細胞面の演算等に用いることができる。この場合、2 次元荷重分布データは、シナプス荷重分布と見なせる。

## 【0 1 1 2】

まず、離散的なコンボリューション演算を行う離散的な 2 次元荷重分布データ  $w(x, y)$  を入力する。この 2 次元荷重分布データは離散的であるので、 $x, y$  は整数とする。この 2 次元荷重分布データによる、2 次元分布データ  $f(x, y)$  に対する離散的コンボリューション演算結果  $Z(x, y)$  は以下の式で表される。

## 【数 3 2】

$$Z(x, y) = \sum_u \sum_v \{w(u, v) \cdot f(u+x, v+y)\}$$

## 【0 1 1 3】

次に、入力された 2 次元荷重分布データ  $w(x, y)$  から、第 2 の実施形態で示した方法と同様の方法により、1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  を生成する。1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  の添え字 1 は基底番号である。ここでの詳細な説明は、第 2 の実施形態で示したものと同様なので説明を割愛する。ここで生成した 1 次元適応化基底組は、コンボリューション演算を行うたびに生成する必要はなく、予め第 2 の実施形態で示した方法により生成しておき、生成した 1 次元適応化基底組を記録して保存しておけばよい。

## 【0 1 1 4】

生成された 1 次元適応化基底組により、2 次元分布データ  $f(x, y)$  に対する離散的コンボリューション演算結果  $Z(x, y)$  は、以下ようになる。

## 【数 3 3】

$$Z(x, y) = \sum_u \left[ \sum_v \{X_1(u) \cdot Y_1(v) \cdot f(u+x, v+y)\} \right]$$

## 【0 1 1 5】

このように、2 次元荷重分布データ  $w(x, y)$  を 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  に分解し、それを用いて離散的コンボリューション演算を行うことで、従来は図 6 に示すように、例えば  $6 \times 6$  の要素で構成される 2 次元荷重分布データによる離散的コン

ポリューション演算であれば、各座標の計 36 の要素に独立に 2 次元荷重分布データの値を与える、つまり各座標に対して独立に 2 次元荷重分布データの値を信号として送るために 36 本の信号線が必要であったが、計 12 本の信号であれば良く、コンポリューション演算器において問題となっていた配線問題を解決することができる。なお、この離散的コンポリューション演算方法を実行するための、積和演算素子をアレイ状に配置したコンポリューション演算器では、図 7 に示すように 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  をアレイの全ての列・行に入力して演算を実行しても良いし、または図 9 に示すように 1 次元適応化基底組  $X_1(x)$ 、 $Y_1(y)$  をそれぞれ一部の列・行にのみ入力して、 $X_1(x)$  と  $Y_1(y)$  の両方が入力されている、アレイ中の一部の領域 (63) においてのみ演算を実行しても構わない。

#### 【0116】

以上、所定の 2 次元荷重分布データによる、2 次元分布データに対する離散的コンポリューション演算において、2 次元荷重分布データを第 2 の実施形態のデータ符号化方法により、1 次元適応化基底組に変換し、それを用いて 2 次元分布データに対する離散的コンポリューション演算を行う方法の例として説明した。

#### 【0117】

以上、本発明に係る実施形態を説明したが、本発明は、図 8 に示すようにコンピュータプログラムによっても実現できることは明らかであるので、本発明はこのようなコンピュータプログラムをもその範疇とする。また、通常コンピュータプログラムは、CDROM 等のコンピュータ可読記憶媒体に記憶されており、それをコンピュータが有する記憶媒体ドライブ装置にセットし、システムにコピーもしくはインストールすることで実行可能になるわけであるから、本発明はかかるコンピュータ可読記憶媒体をもその範疇とするのは明らかである。

#### 【図面の簡単な説明】

#### 【0118】

【図 1】 第 1 の実施形態における符号化処理を示すフローチャートである。

【図 2】 第 2 の実施形態における符号化処理を示すフローチャートである。

【図 3】 第 3 の実施形態における符号化処理を示すフローチャートである。

【図 4】 第 4 の実施形態における符号化処理を示すフローチャートである。

【図 5】 単純な矩形形状の 2 次元画像の例を示す図である。

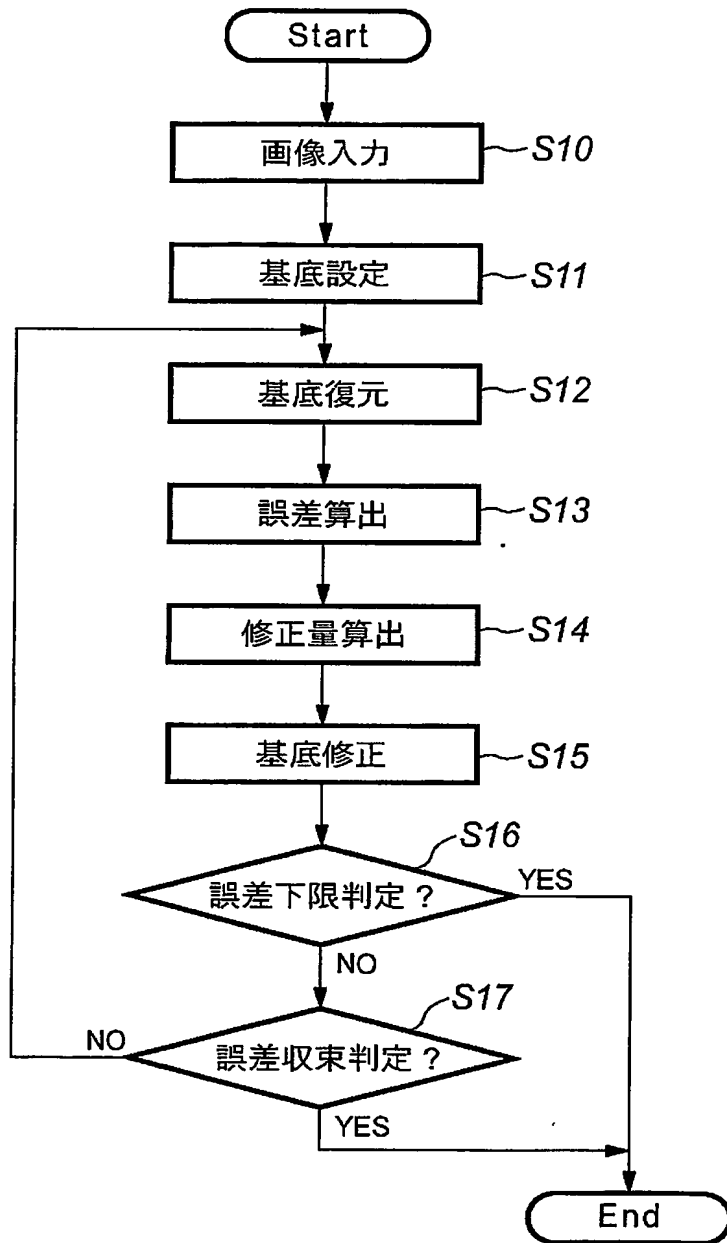
【図 6】 従来の 2 次元荷重分布データの送信構造を示す図である。

【図 7】 第 9 の実施形態の積和演算方法を実行するコンポリューション演算器における、2 次元荷重分布データの送信構造を示す図である。

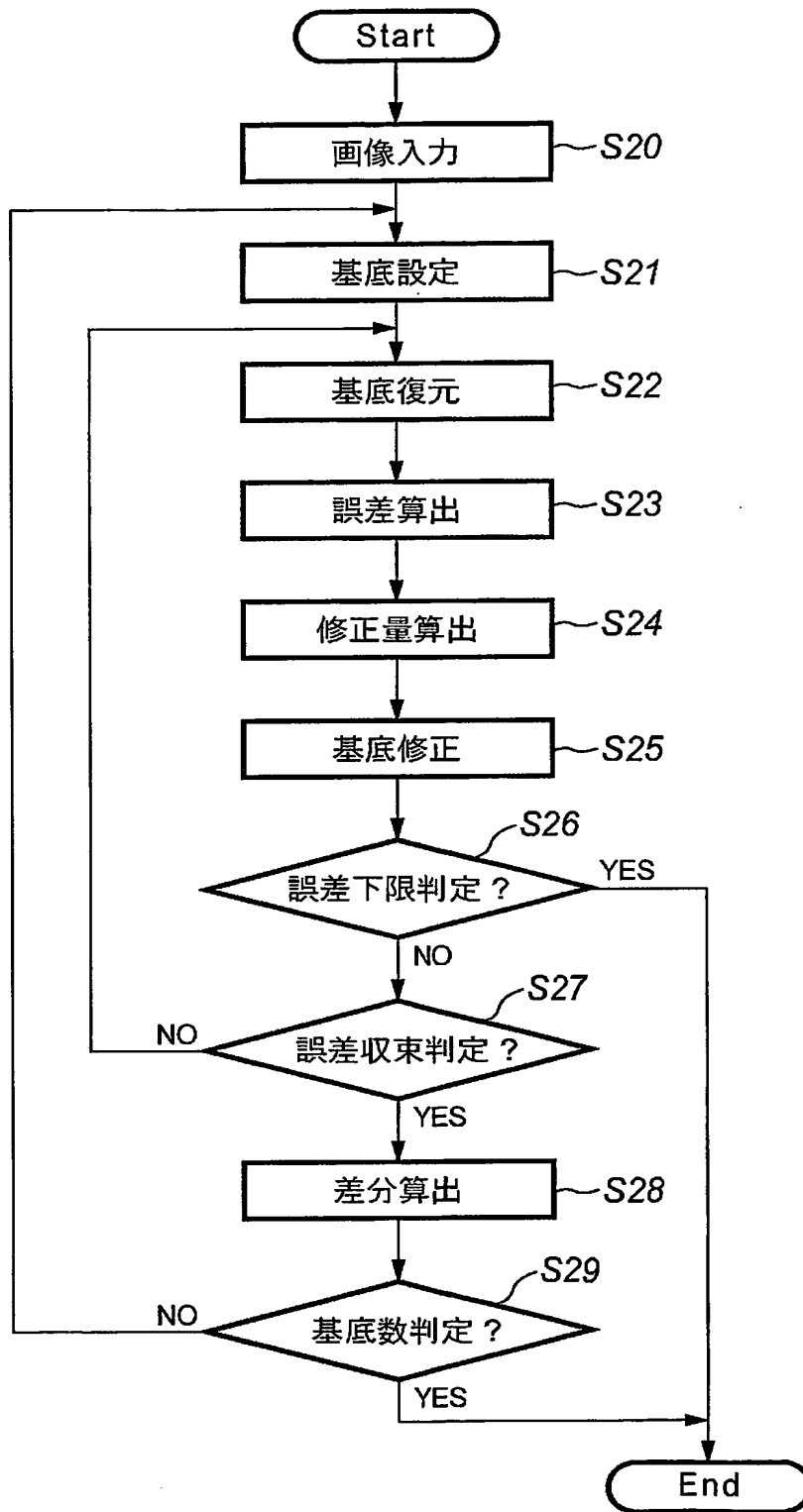
【図 8】 実施形態における装置構成を示すブロック構成図である。

【図 9】 第 9 の実施形態の積和演算方法を実行するコンポリューション演算器における、2 次元荷重分布データの送信構造を示す図である。

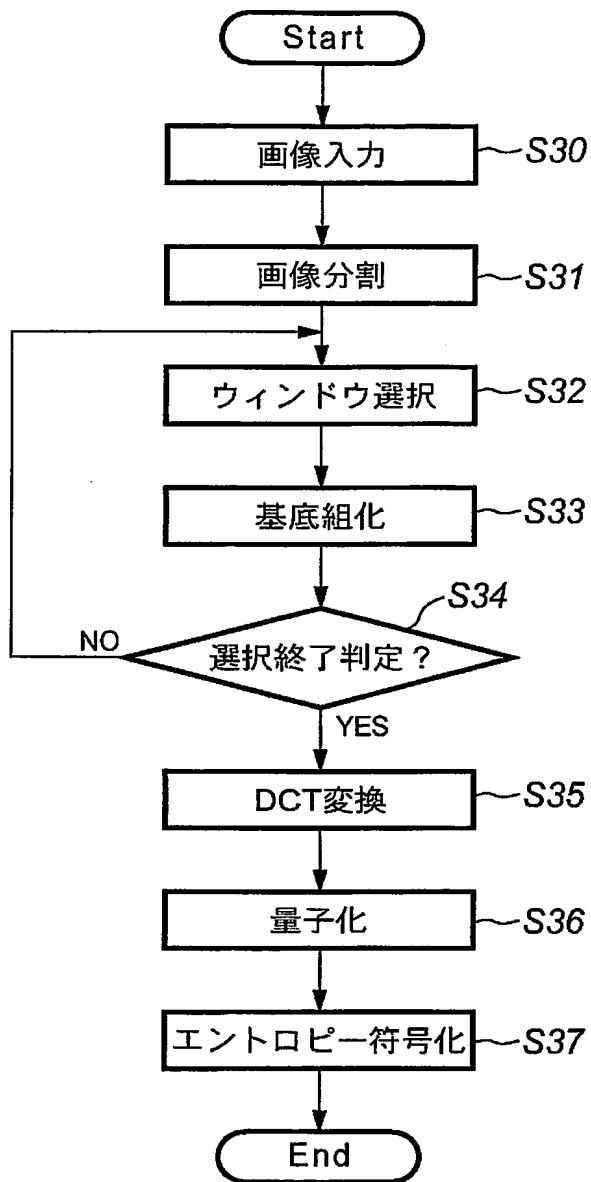
【書類名】 図面  
【図 1】



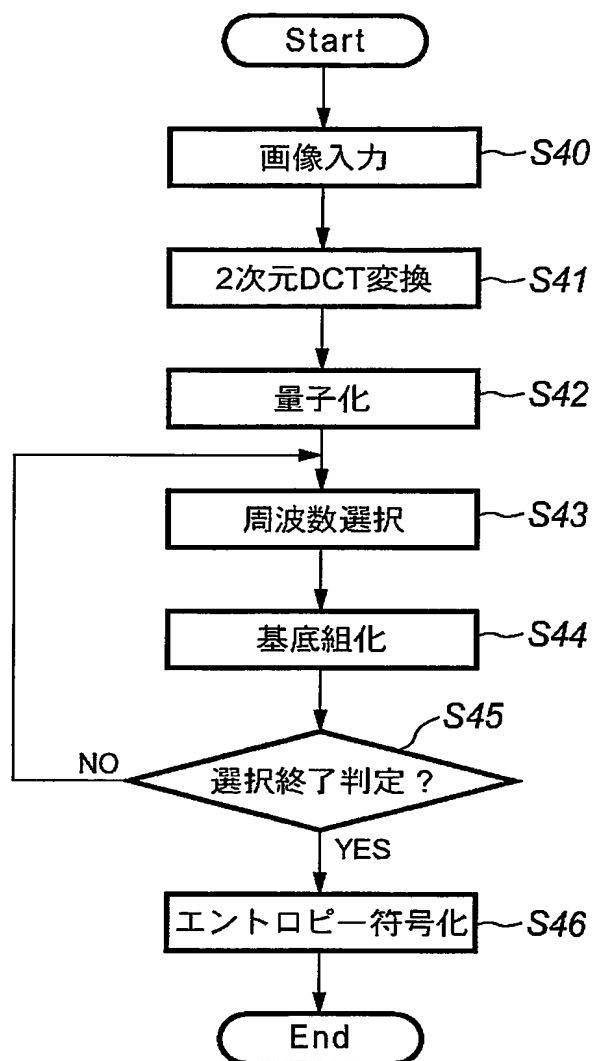
【図 2】



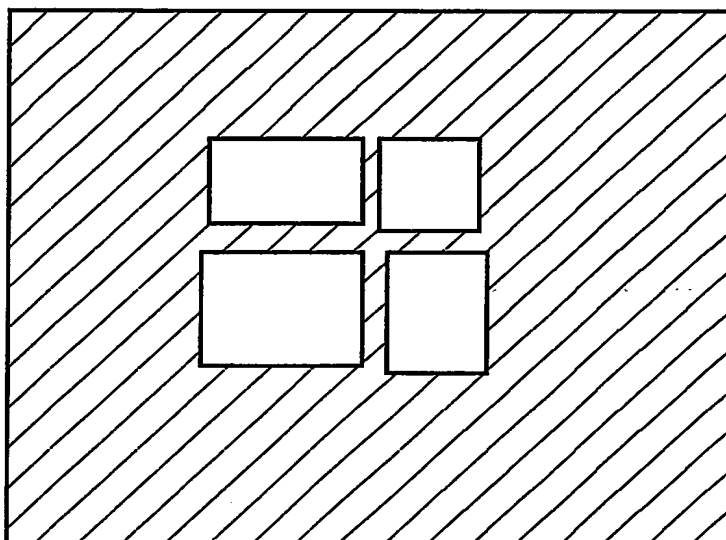
【図 3】



【図 4】

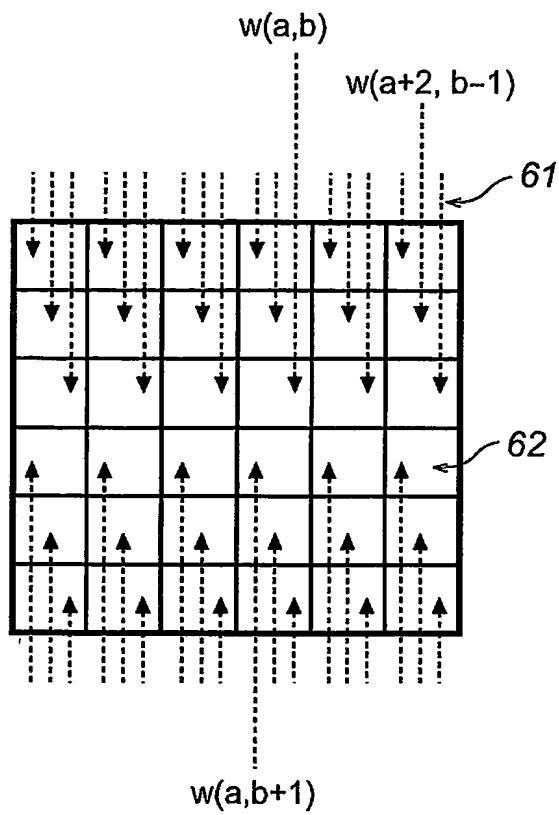


【図 5】

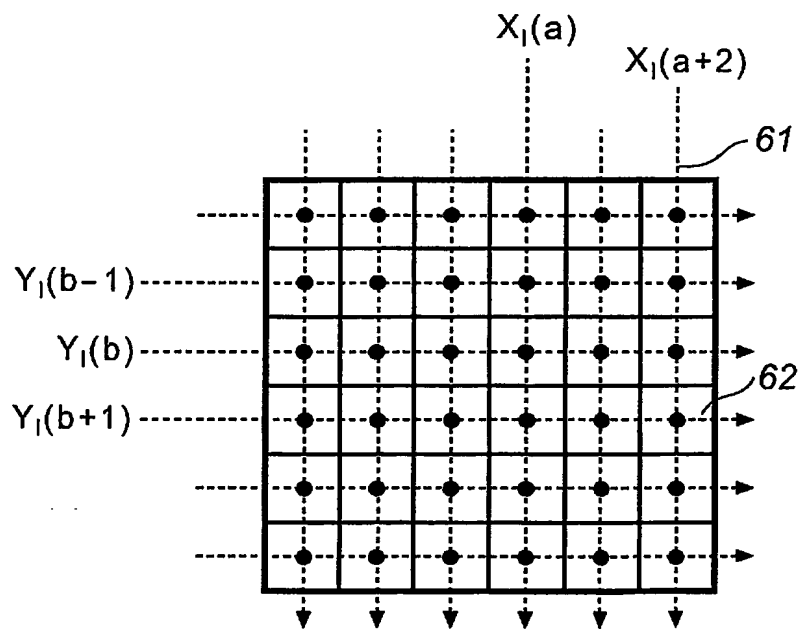




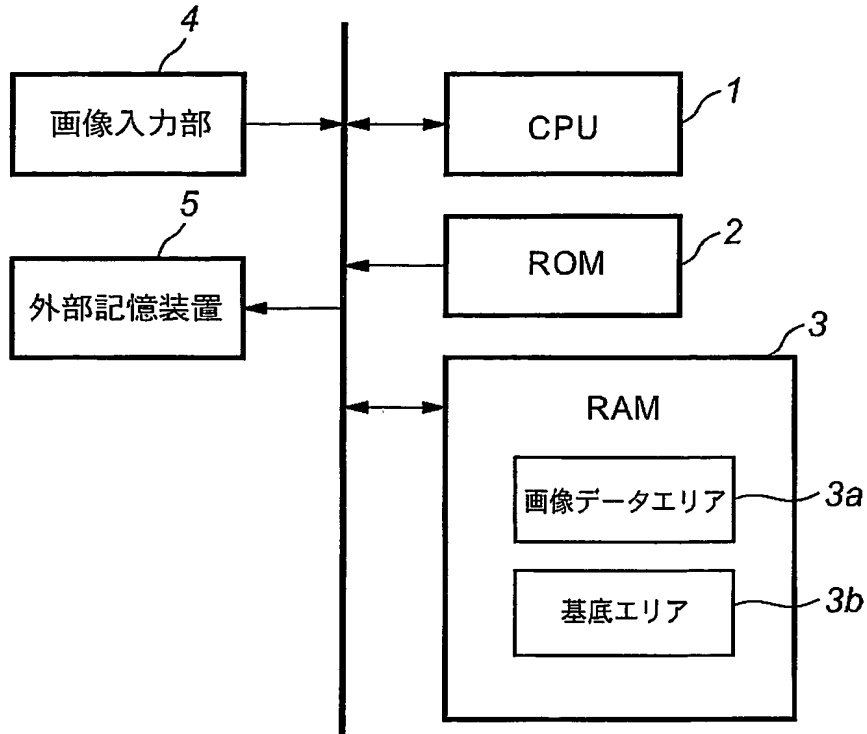
【図 6】



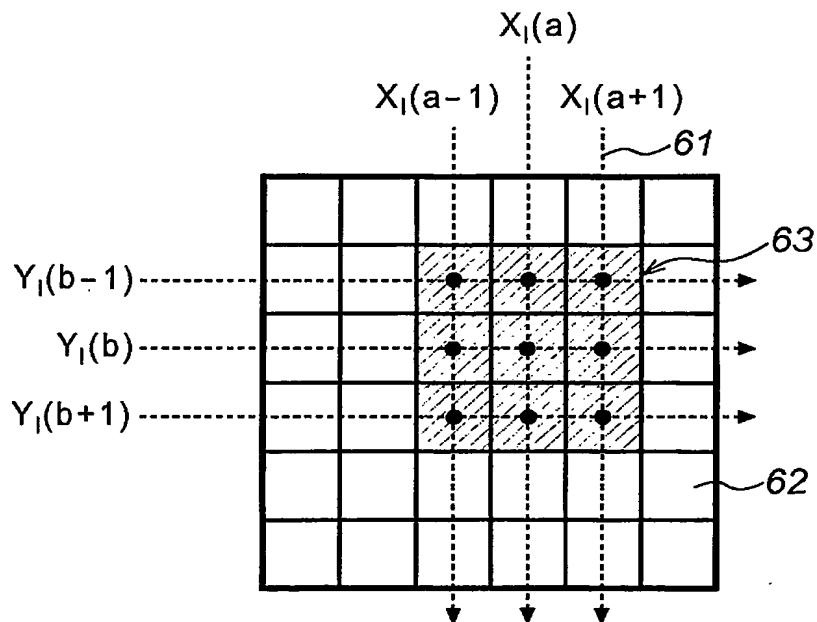
【図 7】



【図 8】



【図 9】



**【書類名】 要約書****【要約】**

**【課題】** 2次元画像等の多次元空間における分布データ等を、高効率に符号化することができる。また2次元のデータを1次元の基底データに分解することが可能になるため、コンポリューション演算器において問題となっていた、2次元並列化による配線の問題を解決する。

**【解決手段】**

符号化しようとする2次元画像データ  $f(x, y)$  を入力 (S10) すると共に、その2次元画像の1次元適応化基底  $X(x)$ 、 $Y(y)$  を確保し、初期値を設定する (S11)。そして、一次元適応化基底に基づき復元画像を生成し (S12)、復元画像と入力画像との誤差  $E$  を演算し (S13)、その誤差  $E$  に基づいて1次元適応化基底  $X(x)$ 、 $Y(y)$  の修正量  $\Delta X(x)$ 、 $\Delta Y(y)$  を求める (S14)。そして、誤差  $E$  が下限に達した (S16)、或いは、収束した (S17) と判断するまで、復元画像の生成と誤差を求めることを行う。

**【選択図】 図1**

【書類名】 手続補正書  
【整理番号】 551414101H  
【提出日】 平成16年 2月 5日  
【あて先】 特許庁長官殿  
【事件の表示】  
【出願番号】 特願2004- 24485  
【補正をする者】  
【識別番号】 000001007  
【氏名又は名称】 キヤノン株式会社  
【代理人】  
【識別番号】 100076428  
【弁理士】  
【氏名又は名称】 大塚 康德  
【電話番号】 03-5276-3241  
【手続補正1】  
【補正対象書類名】 特許願  
【補正対象項目名】 発明者  
【補正方法】 変更  
【補正の内容】  
【発明者】  
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
【氏名】 御手洗 裕輔  
【発明者】  
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
【氏名】 真継 優和  
【発明者】  
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内  
【氏名】 森 克彦  
【発明者】  
【住所又は居所】 広島県東広島市西条下見5丁目10-7 株式会社エイアールテック内  
【氏名】 森江 隆  
【その他】 本件の出願時に、発明者 森江 隆の居所「広島県東広島市西条下見5丁目10-7 株式会社エイアールテック内」を錯誤により、「東京都大田区下丸子3丁目30番2号 キヤノン株式会社内」と誤って記載したため、正しい居所に訂正致したく、ここに補正致します。

認定・付加情報

特許出願の番号	特願 2004-024485
受付番号	50400191812
書類名	手続補正書
担当官	土井 恵子 4264
作成日	平成16年 2月10日

<認定情報・付加情報>

【補正をする者】

【識別番号】	000001007
【住所又は居所】	東京都大田区下丸子3丁目30番2号
【氏名又は名称】	キャノン株式会社

【代理人】 申請人

【識別番号】	100076428
【住所又は居所】	東京都千代田区紀尾井町3番6号 秀和紀尾井町 パークビル7F 大塚国際特許事務所
【氏名又は名称】	大塚 康德

特願 2 0 0 4 - 0 2 4 4 8 5

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 1 0 0 7 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[ 変更理由 ]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/001289

International filing date: 24 January 2005 (24.01.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-024485  
Filing date: 30 January 2004 (30.01.2004)

Date of receipt at the International Bureau: 24 March 2005 (24.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record.**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT\***

☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**